International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.235.0

(09/2005)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Systems aspects

H.323 security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems

ITU-T Recommendation H.235.0

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	**H.230–H.239**
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	H.350–H.359
Quality of service architecture for audiovisual and multimedia services	H.360–H.369
Supplementary services for multimedia	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569
BROADBAND AND TRIPLE-PLAY MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation H.235.0

H.323 security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems

Summary

This Recommendation describes enhancements within the framework of the H.3xx-series Recommendations to incorporate security services such as *Authentication* and *Privacy* (data encryption). The proposed scheme is applicable to both simple point-to-point and multipoint conferences for any terminals which utilize ITU-T Rec. H.245 as a control protocol; also to H.323 systems that use the H.225.0 RAS and/or Call Signalling Protocol.

For example, H.323 systems operate over packet-based networks which do not provide a guaranteed quality of service. For the same technical reasons that the base network does not provide QOS, the network does not provide a secure service. Secure real-time communication over insecure networks generally involves two major areas of concern – *authentication* and *privacy*.

This Recommendation describes the security infrastructure and specific privacy techniques to be employed by the H.3xx-series of multimedia systems. This Recommendation will cover areas of concern for interactive conferencing. These areas include, but are not strictly limited to, authentication and privacy of all real-time media streams that are exchanged in the conference. This Recommendation provides the protocol and algorithms needed between the H.323 entities.

This Recommendation utilizes the general facilities supported in ITU-T Rec. H.245 and as such, any standard which operates in conjunction with this control protocol may use this security framework. It is expected that, wherever possible, other H-series terminals may interoperate and directly utilize the methods described in this Recommendation. This Recommendation will not initially provide for complete implementation in all areas, and will specifically highlight endpoint authentication and media privacy.

This Recommendation includes the ability to negotiate services and functionality in a generic manner, and to be selective concerning cryptographic techniques and capabilities utilized. The specific manner in which they are used relates to systems capabilities, application requirements and specific security policy constraints. This Recommendation supports varied cryptographic algorithms, with varied options appropriate for different purposes; e.g., key lengths. Certain cryptographic algorithms may be allocated to specific security services (e.g., one for fast media stream encryption and another for signalling encryption).

It should also be noted that some of the available cryptographic algorithms or mechanisms may be reserved for export or other national issues (e.g., with restricted key lengths). This Recommendation supports signalling of well-known algorithms in addition to signalling non-standardized or proprietary cryptographic algorithms. There are no specifically mandated algorithms; however, it is strongly suggested that endpoints support as many of the applicable algorithms as possible in order to achieve interoperability. This parallels the concept that the support of ITU-T Rec. H.245 does not guarantee the interoperability between two entities' codecs.

Version 4 of ITU-T Rec. H.235 breaks up the former ITU-T Rec. H.235v3 into a suite of H.235.x subseries Recommendations, and restructures the subseries. New ITU-T Recs H.235.8 and H.235.9 have been added to the suite; other subseries Recommendations have been extended with new functionality (ITU-T Recs H.235.3, H.235.5). ITU-T Rec. H.235.0 holds the H.323 security framework with common text and useful general information for all H.235.x subseries Recommendations.

New H.235.0 Appendices IV, V, and VI provide a mapping of text, figures and tables from ITU-T Rec. H.235 version 3 (2003), including the subsequent Corrigendum 1 and amendments, to the new structure.

Source

ITU-T Recommendation H.235.0 was approved on 13 September 2005 by ITU-T Study Group 16 (2005-2008) under the ITU-T Recommendation A.8 procedure.

Keywords

Authentication, certificate, digital signature, encryption, integrity, key management, multimedia security, security profile.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

ITU-T Recommendation H.235.0

H.323 security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems

1 Scope

The primary purpose of this Recommendation is to provide a security framework for authentication, privacy, and integrity within the current H-series protocol framework. The current text of this Recommendation provides details on implementation with ITU-T Rec. H.323. This framework is expected to operate in conjunction with other H-series protocols that utilize ITU-T Rec. H.245 as their control protocol and/or use the H.225.0 RAS and/or Call Signalling Protocol.

Additional goals in this Recommendation include:

1) Security architecture should be developed as an extensible and flexible framework for implementing a security system for H-series terminals and other H.323-based systems. This should be provided through flexible and independent services and the functionality that they supply. This includes the ability to negotiate and to be selective concerning cryptographic techniques utilized, and the manner in which they are used.
2) Provide security for all communications occurring as a result of H.3xx protocol usage. This includes aspects of connection establishment, call control, and media exchange between all entities. This requirement includes the use of confidential communication (privacy), and may exploit functions for peer authentication as well as protection of the user's environment from attacks.
3) This Recommendation should not preclude integration of other security functions in H.3xx entities which may protect them against attacks from the network.
4) This Recommendation should not limit the ability for any H.3xx-series Recommendation to scale as appropriate. This may include both the number of secured users and the levels of security provided.
5) Where appropriate, all mechanisms and facilities should be provided independent of any underlying transport or topologies. Other means that are outside the scope of this Recommendation may be required to counter such threats.
6) Provisions are made for operation in a mixed environment (secured and unsecured entities).
7) This Recommendation should provide facilities for distributing session keys associated with the cryptography utilized. (This does not imply that public-key-based certificate management must be part of this Recommendation.)
8) This Recommendation provides two security profiles that facilitate interoperability. H.235.1 describes a simple, yet secure password-based security profile while H.235.2 is a signature security profile deploying digital signatures, certificates and a public-key infrastructure that overcomes the limitations of H.235.1.

The security architecture, described in this Recommendation, does not assume that the participants are familiar with each other. It does, however, assume that appropriate precautions have been taken to physically secure the H-series endpoints. The principal security threat to communications, therefore, is assumed to be eavesdropping on the network, or some other method of diverting media streams.

ITU-T Rec. H.323 provides the means to conduct an audio, video and data conference between two or more parties, but does not provide the mechanism to allow each participant to authenticate the identity of the other participants, nor provide the means to make the communications private (i.e., encrypt the streams).

ITU-T Recs H.323, H.324 and H.310 make use of the logical channel signalling procedures of ITU-T Rec. H.245, in which the content of each logical channel is described when the channel is opened. Procedures are provided for expression of receiver and transmitter capabilities, transmissions are limited to what receivers can decode, and receivers may request a particular desired mode from transmitters. The security capabilities of each endpoint are communicated in the same manner as any other communication capability.

Some H-series (H.323) terminals may be used in multipoint configurations. The security mechanism described in this Recommendation will allow for secure operation in these environments, including both centralized and decentralized MCU operation.

1.1 Structure of H.235.x subseries Recommendations

This security framework Recommendation encompasses the following structure within the H.235.x subseries of Recommendations, as shown in Figure 1. This Recommendation contains common text and useful general information for all H.235.x subseries Recommendations.

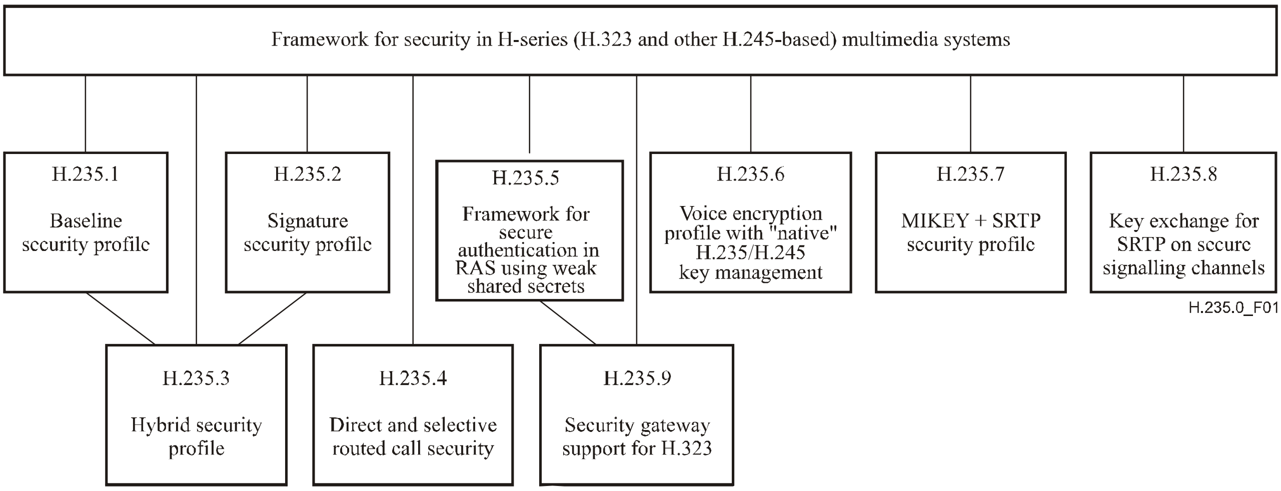

Framework for security in H-series (H.323 and other H.245-based) multimedia systems
H.235.1
Baseline security profile
H.235.2
Signature security profile
H.235.5
Framework for secure authentication in RAS using weak shared secrets
H.235.6
Voice encryption profile with "native" H.235/H.245 key management
H.235.7
MIKEY + SRTP security profile
H.235.8
Key exchange for SRTP on secure signalling channels
H.235.0_F01
H.235.3
Hybrid security profile
H.235.4
Direct and selective routed call security
H.235.9
Security gateway support for H.323

Figure 1/H.235.0 – Structure of H.235.x subseries Recommendations

The vertical lines in Figure 1 indicate direct dependencies from the H.235.0 main text; there may be more indirect dependencies from other H.235.x Recommendations. Several Recommendations could be used in combination and complementarily, see also 6.9.

2 References

2.1 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

– ITU-T Recommendation H.225.0 (2003), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.

- ITU-T Recommendation H.235 (2003), *Security and encryption for H-series (H.323 and other H.245-based) multimedia terminals* plus Amendment 1 (2004), plus Corrigendum 1 (2005).
- ITU-T Recommendation H.235.1 (2005), *H.323 security: Baseline security profile*.
- ITU-T Recommendation H.235.2 (2005), *H.323 security: Signature security profile*.
- ITU-T Recommendation H.235.3 (2005), *H.323 security: Hybrid security profile*.
- ITU-T Recommendation H.235.4 (2005), *H.323 security: Direct and selective routed call security*.
- ITU-T Recommendation H.235.5 (2005), *H.323 security: Framework for secure authentication in RAS using weak shared secrets*.
- ITU-T Recommendation H.235.6 (2005), *H.323 security: Voice encryption profile with native H.235/H.245 key management*.
- ITU-T Recommendation H.235.7 (2005), *H.323 security: Usage of the MIKEY key management protocol for the Secure Real Time Transport Protocol (SRTP) within H.235*.
- ITU-T Recommendation H.235.8 (2005), *H.323 security: Key exchange for SRTP using secure signalling channels*.
- ITU-T Recommendation H.235.9 (2005), *H.323 security: Security gateway support for H.323*.
- ITU-T Recommendation H.245 (2005), *Control protocol for multimedia communication*.
- ITU-T Recommendation H.323 (2003), *Packet-based multimedia communications systems*.
- ITU-T Recommendation H.530 (2002), *Symmetric security procedures for H.323 mobility in H.510,* plus Corrigendum 1 (2003).
- ITU-T Recommendation Q.931 (1998), *ISDN user-network interface layer 3 specification for basic call control*.
- ITU-T Recommendation X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications*.

  ISO 7498-2:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture*.
- ITU-T Recommendation X.803 (1994) | ISO/IEC 10745:1995, *Information technology – Open Systems Interconnection – Upper layers security model*.
- ITU-T Recommendation X.810 (1995) | ISO/IEC 10181-1:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Overview*.
- ITU-T Recommendation X.811 (1995) | ISO/IEC 10181-2:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Authentication framework*.
- ISO/IEC 9798-2:1999, *Information technology – Security techniques – Entity authentication – Part 2: Mechanisms using symmetric encipherment algorithms*.
- ISO/IEC 9798-3:1998, *Information technology – Security techniques – Entity authentication – Part 3: Mechanism using digital signature techniques*.
- ISO/IEC 9798-4:1999, *Information technology – Security techniques – Entity authentication – Part 4: Mechanisms using a cryptographic check function*.
- ISO/IEC 15946-1:2002, *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 1: General*.

- ISO/IEC 15946-2:2002, *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 2: Digital signatures*.
- ATM Forum: af-sec-0100.002 (2001), *ATM Security Specification Version 1.1*.
- IETF RFC 2246 (1999), *The TLS Protocol Version 1.0*.
- IETF RFC 2401 (1998), *Security Architecture for the Internet Protocol*.
- IETF RFC 2407 (1998), *The Internet IP Security Domain of Interpretation for ISAKMP*.
- IETF RFC 2408 (1998), *Internet Security Association and Key Management Protocol (ISAKMP)*.
- IETF RFC 2865 (2000), *Remote Authentication Dial In User Service (RADIUS)*.
- IETF RFC 3546 (2003), Transport Layer Security Protocol (TLS) Extensions.
- IETF RFC 3830 (2004), *MIKEY: Multimedia Internet KEYing*.

2.2 Informative references

[Daemon]	DAEMON (J.), *Cipher and Hash function design*, Ph.D. Thesis, Katholieke Universiteit Leuven, March 1995.
[ESP]	IETF RFC 2406 (1998), *IP Encapsulating Security Payload (ESP)*.
[OAKLEY]	IETF RFC 2412 (1998), *The OAKLEY Key Determination Protocol*.
[IKE]	IETF RFC 2409 (1998), *The Internet Key Exchange (IKE)*.
[ISO|IEC 14888-3]	ISO/IEC 14888-3:1998, *Information technology – Security techniques – Digital signatures with appendix; Part 3: Certificate-based mechanisms*.
[J.170]	ITU-T Recommendation J.170 (2005), *IPCablecom security specification*.
[RTP]	IETF RFC 3550 (2003), *RTP: A transport Protocol for Real-Time Applications*.
[Schneier]	SCHNEIER (B.), *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd Edition, John Wiley & Sons, Inc., 1995.
[SRTP]	IETF RFC 3711 (2004), *The Secure Real-time Transport Protocol (SRTP)*.

3 Terms and definitions

For the purposes of this Recommendation, the definitions given in clauses 3/H.323, 3/H.225.0 and 3/H.245 apply along with those in this clause. Some of the following terms used in this Recommendation are also defined in ITU-T Recs X.800 | ISO 7498-2, X.803 | ISO/IEC 10745, X.810 | ISO/IEC 10181-1 and X.811 | ISO/IEC 10181-2.

3.1 **access control**: The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner (ITU-T Rec. X.800).

3.2 **authentication**: The provision of assurance of the claimed identity of an entity (ITU-T Rec. X.811 | ISO/IEC 10181-2).

3.3 **authorization**: The granting of permission on the basis of authenticated identification.

3.4 attack: The activities undertaken to bypass or exploit deficiencies in a system's security mechanisms. By a direct attack on a system they exploit deficiencies in the underlying algorithms, principles, or properties of a security mechanism. Indirect attacks are performed when they bypass the mechanism, or when they make the system use the mechanism incorrectly.

3.5 certificate: A set of security-relevant data issued by a security authority or trusted third party, together with security information which is used to provide the integrity and data origin authentication services for the data (ITU-T Rec. X.810 | ISO/IEC 10181-1). In this Recommendation, the term refers to "public key" certificates which are values that represent an owner's public key (and other optional information) as verified and signed by a trusted authority in an unforgeable format.

3.6 cipher: A cryptographic algorithm, a mathematical transform.

3.7 confidentiality: The property that prevents disclosure of information to unauthorized individuals, entities, or processes.

3.8 cryptographic algorithm: Mathematical function that computes a result from one or several input values.

3.8 *bis* EC-GDSA: Elliptic curve digital signature with appendix analog of the NIST Digital Signature Algorithm (DSA) (see also ISO/IEC 15946-2, chapter 5).

3.8 *ter* elliptic Curve Cryptosystem: A public-key cryptosystem (see section 8.7 of *ATM Forum Security Specification Version 1.1*).

3.8 *quat* elliptic Curve Key Agreement Scheme – Diffie-Hellman: The Diffie-Hellman key agreement scheme using elliptic curve cryptography.

3.9 encipherment: Encipherment (encryption) is the process of making data unreadable to unauthorized entities by applying a cryptographic algorithm (an encryption algorithm). Decipherment (decryption) is the reverse operation by which ciphertext is transformed to plaintext.

3.10 integrity: The property that data has not been altered in an unauthorized manner.

3.11 key management: The generation, storage, distribution, deletion, archiving and application of keys in accordance with a security policy (ITU-T Rec. X.800).

3.12 media stream: A media stream can be of type audio, video or data or a combination of any of them. Media stream data conveys user or application data (payload) but no control data.

3.13 non-repudiation: Protection from denial by one of the entities involved in a communication of having participated in all or part of the communication.

3.14 privacy: A mode of communication in which only the explicitly enabled parties can interpret the communication. This is typically achieved by encryption and shared key(s) for the cipher.

3.15 private channel: For this Recommendation, a private channel is one that is a result of prior negotiation on a secure channel. In this context, it may be used to handle media streams.

3.16 public key cryptography: An encryption system utilizing asymmetric keys (for encryption/decryption) in which the keys have a mathematical relationship to each other which cannot be reasonably calculated.

3.17 security profile: A (sub)set of consistent, interoperable procedures and features out of ITU-T Rec. H.235 useful for securing H.323 multimedia communication among the involved entities in a specific scenario.

3.18 spamming: A denial-of-service attack when sending unauthorized data in excess to a system. A special case is media spamming when sending RTP packets on UDP ports. Usually the system is flooded with packets; the processing consumes precious system resources.

3.19 symmetric (secret-key based) cryptographic algorithm: An algorithm for performing encipherment or the corresponding algorithm for performing decipherment in which the same key is required for both encipherment and decipherment (ITU-T Rec. X.810 | ISO/IEC 10181-1).

3.20 threat: A potential violation of security (ITU-T Rec. X.800 | ISO 7498-2).

4 Symbols and abbreviations

This Recommendation uses the following abbreviations:

X || Y	Concatenation of X and Y
3DES	Triple DES
AES	Advanced Encryption Algorithm
ALG	Application Layer Gateway
ASN.1	Abstract Syntax Notation One
BES	Back-end Server
CA	Certificate Authority
CBC	Cipher Block Chaining
CFB	Cipher Feedback mode
CRL	Certificate Revocation List
DES	Data Encryption Standard
DH	Diffie-Hellman
DNS	Domain Name System
DSS	Digital Signature Standard
DTMF	Dual Tone Multi-Frequency
ECB	Electronic Code Book
ECC and EC	Elliptic Curve Cryptosystem (see section 8.7 of *ATM Forum Security Specification Version 1.1*). A public-key cryptosystem.
EC-GDSA	Elliptic curve digital signature with appendix analog of the NIST Digital Signature Algorithm (DSA) (see also ISO/IEC 15946-2, chapter 5)
ECKAS-DH	Elliptic Curve Key Agreement Scheme – Diffie-Hellman. The Diffie-Hellman key agreement scheme using elliptic curve cryptography
EOFB	Enhanced OFB mode
EP	Endpoint
GK	Gatekeeper
GW	Gateway
ICV	Integrity Check Value
ID	Identifier
IETF	Internet Engineering Task Force
IPsec	Internet Protocol Security
ISAKMP	Internet Security Association Key Management Protocol

ISO	International Organization for Standardization
IV	Initialization Vector
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
MC	Multicast Controller
MCU	Multipoint Control Unit
MPS	Multiple Payload Stream
NAT	Network Address Translation
OCSP	Online Certificate Status Protocol
OFB	Output Feedback Mode
OID	Object Identifier
PDU	Protocol Data Unit
PKI	Public Key Infrastructure
POTS	Plain Old Telephone Service
PRF	Pseudo-Random Function
Q&A	Question and Answer
QoS	Quality of Service
RAS	Registration, Admission, Status
RSA	Rivest, Shamir and Adleman (public key algorithm)
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
SASET	Secure Audio Simple Endpoint Type
SDU	Service Data Unit
SHA1	Secure Hash Algorithm 1
SRTP	Secure Real-Time Transport Protocol
SSL	Secure Socket Layer
TLS	Transport Level Security
TSAP	Transport Service Access Point
TTP	Trusted Third Party
UDP	User Datagram Protocol
XOR, ⊕	Exclusive OR

5 Conventions

In this Recommendation the following conventions are used:

– "shall" indicates a mandatory requirement.

– "should" indicates a suggested but optional course of action.

– "may" indicates an optional course of action rather than a recommendation that something take place.

References to clauses, subclauses, annexes and appendices refer to those items within this Recommendation unless another Recommendation is explicitly listed. For example, "1.4" refers to clause 1.4 of this Recommendation; "6.4/H.245" refers to clause 6.4 in ITU-T Rec. H.245.

This Recommendation describes the use of "n" different message types: H.245, RAS, Q.931, etc. To distinguish between the different message types, the following convention is followed. H.245 message and parameter names consist of multiple concatenated words highlighted in bold typeface (**maximumDelayJitter**). RAS message names are represented by three-letter abbreviations (**ARQ**). Q.931 message names consist of one or two words with the first letters capitalized (**Call Proceeding**).

This Recommendation uses the notion of setting a compound ASN.1 data structure to NULL; for example, "**paramS** set to NULL" (see clause 7/H.235.1, clause 8/H.235.1, 9.1/H.235.1, 9.2/H.235.1, clause 7/H.235.2, clause 9/H.235.2, 15.1/H.235.2 and 15.2/H.235.2). This shall mean that all optional elements in the particular SEQUENCE (i.e., **Params**) are absent.

This Recommendation defines various object identifiers (OIDs) for signalling security capabilities, procedures or security algorithms. These OIDs relate to a hierarchical tree of assigned values that may originate from external sources, or are part of the ITU-T maintained OID tree. Those OIDs that are specifically related to ITU-T Rec. H.235 have the following appearance in the text:

"OID" = {itu-t (0) recommendation (0) h (8) 235 version (0) **V** **N**} where **V** symbolically represents a single decimal digit denoting the corresponding version of ITU-T Rec. H.235; e.g., 1, 2, 3 or 4. **N** symbolically represents a decimal number uniquely identifying the instance of the OID and thus, the procedure, algorithm or security capability.

Thus, the ASN.1 encoded OID consists of a sequence of numbers. For convenience, a textual mnemonic shorthand string notation for each OID is used in the text such as "OID". A mapping is given that relates each OID string with the ASN.1 sequence of numbers. Implementations conforming to ITU-T Rec. H.235 shall use only the ASN.1 encoded numbers.

6 System introduction

Figure 2 gives an overview of the scope of this Recommendation within ITU-T Rec. H.323.

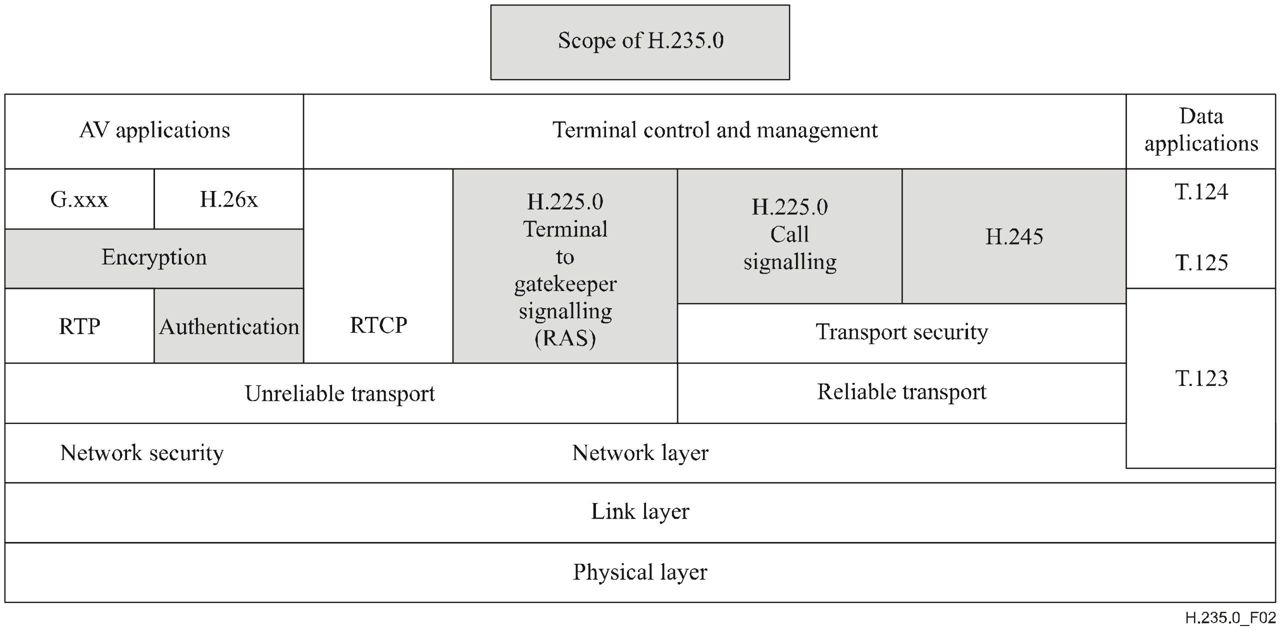


Figure 2/H.235.0 – Overview

For ITU-T Rec. H.323, the signalling of usage of TLS (RFC 2246, RFC 3546), IPsec or a proprietary mechanism on the H.245 control channel shall occur on the secured or unsecured H.225.0 channel during the initial Q.931 message exchange.

6.1 Summary

1) The call signalling channel may be secured using TLS (RFC 2246, RFC 3546) or IPsec (RFC 2401, [ESP]) on a secure well-known port (ITU-T Rec. H.225.0).
2) Users may be authenticated either during the initial call connection, in the process of securing the H.245 channel and/or by exchanging certificates on the H.245 channel.
3) The encryption capabilities of a media channel are determined by extensions to the existing capability negotiation mechanism.
4) Initial distribution of key material from the master is via H.245 **OpenLogicalChannel** or **OpenLogicalChannelAck** messages.
5) Re-keying may be accomplished by H.245 commands: **EncryptionUpdateCommand**, **EncryptionUpdateRequest, EncryptionUpdate and EncryptionUpdateAck**.
6) Key material distribution is protected either by operating the H.245 channel as a private channel, or by specifically protecting the key material using the selected exchanged certificates.
7) The security protocols presented conform either to ISO published standards or to IETF proposed standards.

6.2 Authentication

The process of authentication verifies that the respondents are, in fact, who they say they are. Authentication may be accomplished in conjunction with the exchange of public key-based certificates. Authentication may also be accomplished by an exchange which utilizes a shared secret between the entities involved. This may be a static password or some other *a priori* piece of information.

This Recommendation describes the protocol for exchanging the certificates, but does not specify the criteria by which they are mutually verified and accepted. In general, certificates give some assurance to the verifier that the presenter of the certificate is who he says he is. The intent behind the certificate exchange is to authenticate the *user* of the endpoint, not simply the physical endpoint. Using digital certificates, an authentication protocol proves that the respondents possess the private keys corresponding to the public keys contained in the certificates. This authentication protects against man-in-the-middle attacks, but does not automatically prove who the respondents are. To do this normally requires that there be some policy regarding the other contents of the certificates. For authorization certificates, for example, the certificate would normally contain the service-provider's identification along with some form of user account identification prescribed by the service provider.

The authentication framework in this Recommendation does not prescribe the contents of certificates (i.e., does not specify a certificate policy) beyond that required by the authentication protocol. However, an application using this framework may impose high-level policy requirements such as presenting the certificate to the user for approval. This higher level policy may either be automated within the application or require human interaction.

For authentication which does not utilize digital certificates, this Recommendation provides the signalling to complete various challenge/response scenarios. This method of authentication requires prior coordination by the communicating entities so that a shared secret may be obtained. An example of this method would be a customer of a subscription-based service.

As a third option, the authentication may be completed within the context of a separate security protocol such as TLS (RFC 2246, RFC 3546) or RFC 2409 [IKE].

Both bidirectional and unidirectional authentication may be supported by peer entities. This authentication may occur on some or all of the communication channels.

All of the specific authentication mechanisms described in this Recommendation are identical to, or derived from, ISO-developed algorithms as specified in Parts 2 to 3 of ISO/IEC 9798, or based on IETF protocols.

6.2.1 Certificates

The standardization of certificates, including their generation, administration and distribution is outside the scope of this Recommendation. The certificates used to establish secure channels (call signalling and/or call control) shall conform to those prescribed by whichever protocol has been negotiated to secure the channel.

It should be noted that for authentication utilizing public key certificates, the endpoints are required to provide digital signatures using the associated private key value. The exchange of public key certificates alone does not protect against man-in-the-middle attacks. The H.235 protocols conform to this requirement.

6.3 Call establishment security

There are at least two reasons to motivate securing the call establishment channel (e.g., H.323 using Q.931). The first is for simple authentication, before accepting the call. The second reason is to allow for call authorization. If this functionality is desired in the H-series terminal, a secure mode of communication should be used (such as TLS/IPsec for H.323) before the exchange of call connection messages. Alternatively, the authorization may be provided based upon a service-specific authentication. The constraints of a service-specific authorization policy are outside the scope of this Recommendation.

6.4 Call control (H.245) security

The call control channel (H.245) should also be secured in some manner to provide for subsequent media privacy. The H.245 channel shall be secured using any negotiated privacy mechanism (this includes the option of "none"). H.245 messages are utilized to signal encryption algorithms and encryption keys used in the shared, private, media channels. The ability to do this, on a logical channel by logical channel basis, allows different media channels to be encrypted by different mechanisms. For example, in centralized multipoint conferences, different keys may be used for streams to each endpoint. This may allow media streams to be made private for each endpoint in the conference. In order to utilize the H.245 messages in a secure manner, the entire H.245 channel (logical channel 0) should be opened in a negotiated secure manner.

The mechanism by which H.245 is made secure is dependent on the H-series terminals involved. The only requirement on all systems that utilize this security structure is that each shall have some manner in which to negotiate and/or signal that the H.245 channel is to be operated in a particular secured manner before it is actually initiated. For example, H.323 will utilize the H.225.0 connection signalling messages to accomplish this.

6.5 Media stream privacy

This Recommendation describes media privacy for media streams carried on packet-based transports. These channels may be unidirectional with respect to H.245 logical channel characterizations. The channels are not required to be unidirectional on a physical or transport level.

A first step in attaining media privacy should be the provision of a private control channel on which to establish cryptographic keying material and/or set up the logical channels which will carry the encrypted media streams. For this purpose, when operating in a secure conference, any participating endpoints may utilize an encrypted H.245 channel. In this manner, cryptographic algorithm selection and encryption keys as passed in the H.245 **OpenLogicalChannel** command are protected.

The H.245 secure channel may be operated with characteristics different from those in the private media channel(s) as long as it provides a mutually acceptable level of privacy. This allows for the security mechanisms protecting media streams and any control channels to operate in a completely independent manner, providing completely different levels of strength and complexity.

If it is required that the H.245 channel be operated in a non-encrypted manner, the specific media encryption keys may be encrypted separately in the manner signalled and agreed to by the participating parties. A logical channel of type **h235Control** may be utilized to provide the material to protect the media encryption keys. This logical channel may be operated in any appropriately negotiated mode.

The privacy (encryption) of data carried in logical channels shall be in the form specified by the **OpenLogicalChannel**. Transport-specific header information shall not be encrypted. The privacy of data is to be based upon end-to-end encryption.

6.6 Trusted elements

The basis for authentication (trust) and privacy is defined by the terminals of the communications channel. For a connection establishment channel, this may be between the caller and a hosting network component. For example, a telephone "trusts" that the network switch will connect it with the telephone whose number has been dialled. For this reason, any entity which terminates an encrypted H.245 control channel or any **encryptedData** type logical channels shall be considered a trusted element of the connection; this may include MC(U)s and gateways. The result of trusting an element is the confidence to reveal the privacy mechanism (algorithm and key) to that element.

Given the above, it is incumbent upon participants in the communications path to authenticate any and all "trusted" elements. This will normally be done by certificate exchange as would occur for the "standard" end-to-end authentication. This Recommendation will not require any specific level of authentication, other than to suggest that it be acceptable to all entities using the trusted element. Details of a trust model and certificate policy are for further study.

Privacy can be assured between the two endpoints only if connections between trusted elements are proven to be protected against man-in-the-middle attacks.

6.6.1 Key escrow

Although not specifically required for operation, this Recommendation contains provision for entities utilizing the H.235 protocol to support the facility known as trusted third party (TTP) within the signalling elements.

The ability to recover lost media encryption keys should be supported in installations where this functionality is desired or required.

Key escrow is a facility which is often referred to as a Trusted Third Party (TTP). This facility is for further study.

6.7 Non-repudiation

For further study.

6.8 Mobility security

H.323-based systems may be deployed in a mobility environment according to ITU-T Rec. H.510. Security procedures and protocols for such systems are described in ITU-T Rec. H.530. ITU-T Rec. H.530 deploys protocols and procedures from this Recommendation.

6.9 Security profiles

This Recommendation references a couple of security profiles of H.235 (i.e., H.235.1, H.235.2, H.235.3, H.235.4, H.235.5, H.235.6, H.235.7, H.235.8 and H.235.9). A security profile specifies specific usage of H.235 or a subset of H.235 functionality for well-defined environments with scoped applicability.

Depending on the environment and application, security profiles may be implemented either selectively or altogether. Typically, H.235-enabled systems indicate within object identifiers as part of signalling messages which security profiles they deploy. H.235-enabled systems should choose the security profile according to their needs.

Optionally, endpoints may initially offer multiple security profiles simultaneously, in **RRQ**/**GRQ** messages, and let the gatekeeper select the most adequate one by answering it in the **RCF**/**GCF** message. **LRQ**/**LCF** transactions between gatekeepers may also carry several security profiles. When calculating digital signatures or hash values to provide message integrity, first the hash values and digital signatures which do not provide message integrity should be calculated over the field subset and set in the message, all the digital signatures and hash values that provide message integrity should be set to zeroes in the message buffer, then all the digital signatures and hash values should be calculated using this buffer, and then set in the message.

Each of the subseries Recommendations is written as a security profile of H.235.0. A security profile of H.235.0 typically comprises a use-case specific instantiation of H.235.0 within a particular scenario and/or holds a particular security feature specification or a combination of security mechanisms/security profiles.

All security profiles are optional within H.235.0.

Figure 3 illustrates some typical and possible combinations of security profiles. A straight line indicates that a pairwise combination of security profiles is defined and possible. A dashed line indicates that a combination is generally possible yet such a combination may not be very useful. Missing lines indicate that a particular combination is not yet defined.

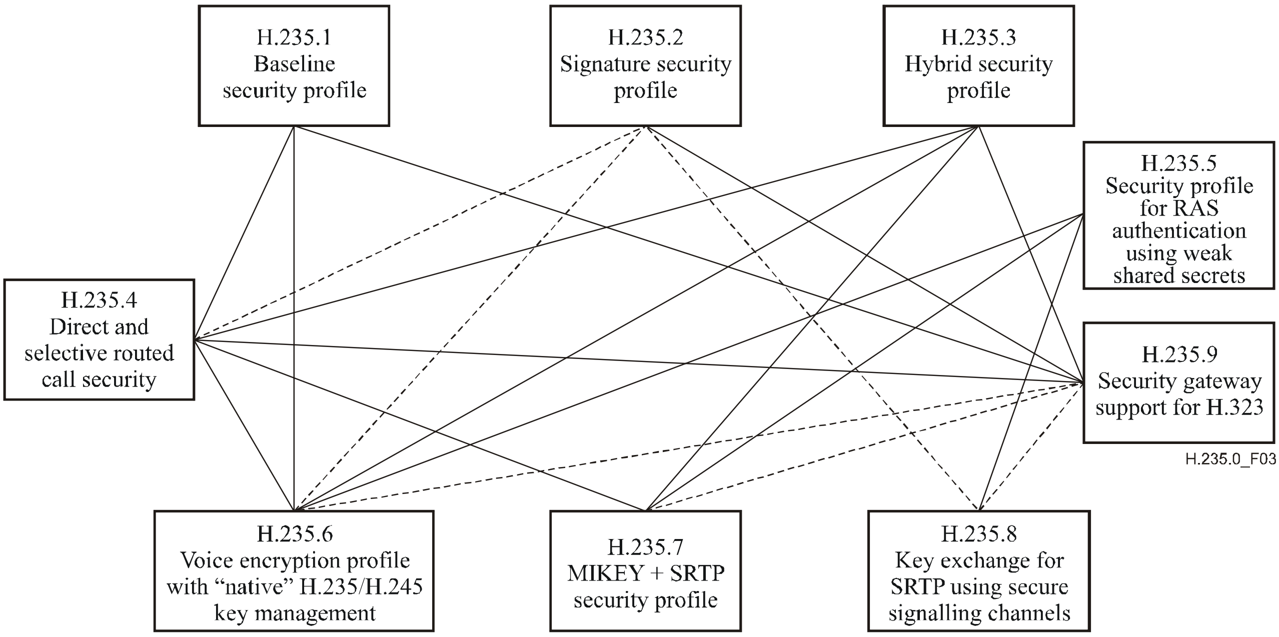


Figure 3/H.235.0 – Illustration of security profile combinations

6.10 Secured NAT/firewall traversal

ITU-T Rec. H.235.9 specifies procedures on how to discover the presence of Security Gateways (such as ALGs) in the H.225.0 RAS signalling path between H.323 entities (Gatekeeper, endpoint) and how a Gatekeeper and a Security Gateway share security information in order to preserve signalling integrity and privacy.

ITU-T Recs H.235.1 (Procedure IA) and H.235.2 (Authentication-Only procedure) offer complementary specific procedures that allow H.235-based message authentication of H.225.0 RAS and call signalling protocols to traverse NAT/Firewall devices.

7 Connection establishment procedures

As stated in the system introduction clause, both the call connection channel (H.225.0 for H.323-series) and call control (H.245) channel shall operate in the negotiated secured or unsecured mode starting with the first exchange. For the call connection channel, this is done *a priori* (for H.323, a TLS secured TSAP (port 1300) shall be utilized for the Q.931 messages). For the call control channel, security mode is determined by information passed in the initial connection setup protocol in use by the H-series terminal.

In the cases in which there are no overlapping security capabilities, the called terminal may refuse the connection. The error returned should convey no information about any security mismatch; the calling terminal will have to determine the problem by some other means. In cases where the calling terminal receives a message without sufficient security capabilities, it should terminate the call.

If the calling and called terminals have compatible security capabilities, it shall be assumed by both sides that the H.245 channel shall operate in the secure mode negotiated. Failure to set up the H.245 channel in the secure mode determined here should be considered a protocol error and the connection terminated.

ITU-T Rec. H.235.6 provides further security connection establishment procedures including key management; see clauses 7 and 8/H.235.6.

8 Authentication signalling and procedures

Authentication is, in general, based either on using a shared secret (you are authenticated properly if you know the secret) or on public key-based methods with certifications (you prove your identity by possessing the correct private key). A shared secret and the subsequent use of symmetric cryptography require a prior contact between the communicating entities. A prior face-to-face or secure contact can be replaced by generating or exchanging the shared secret key with methods based on public key cryptography, e.g., by Diffie-Hellman key exchange. The communication parties in the key generation and exchange have to be authenticated, for example, by using digitally signed messages; otherwise the communication parties cannot be sure with whom they share the secret.

This Recommendation presents authentication methods based on subscription, i.e., there must be a prior contact for sharing a secret and authentication methods where public key cryptography is directly used in authentication, or it is used for generating the shared secret.

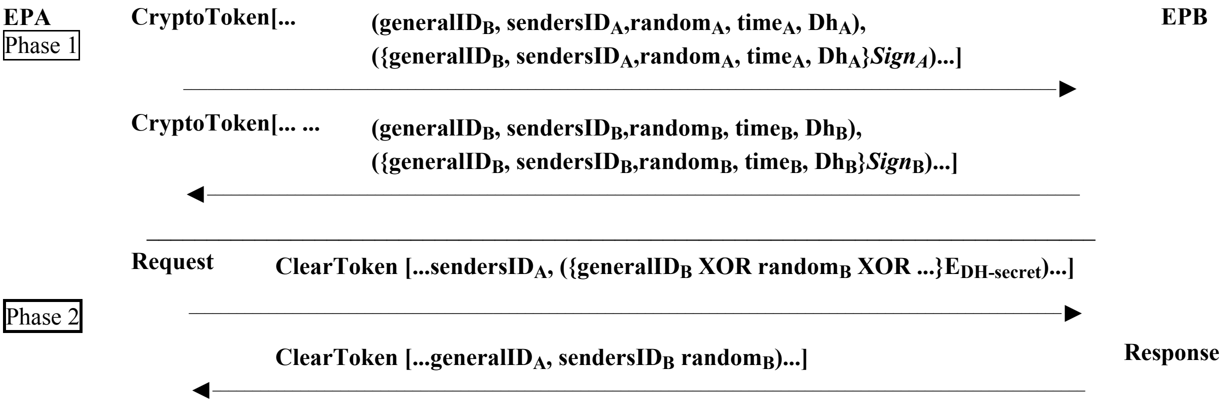
8.1 Diffie-Hellman with optional authentication

The intent is not to provide absolute, user-level authentication. This method provides signalling to generate a shared secret between two entities which may lead to keying material for private communications.

At the end of this exchange, both the entities will possess a shared secret key along with a chosen algorithm with which to utilize this key. This shared secret key may now be used on any subsequent request/response exchanges. It should be noted that in rare cases, the Diffie-Hellman exchange may generate known *weak* keys for particular algorithms. When this is the case, either entity should disconnect and reconnect to establish a new key set.

The first phase of Figure 4 demonstrates the data exchanged during the Diffie-Hellman. The second phase allows for application- or protocol-specific request messages to be authenticated by the responder. Note that a new random value may be returned with each response.

NOTE – If the messages are exchanged over an insecure channel, then digital signatures (or other message origin authentication method) must be used in order to authenticate the parties between whom the secret will be shared. An optional signature element may also be provided; these are illustrated in ***italics*** below.


EPA
Phase 1
CryptoToken[...
(generalID$_B$, sendersID$_A$,random$_A$, time$_A$, Dh$_A$),
({generalID$_B$, sendersID$_A$,random$_A$, time$_A$, Dh$_A$}*Sign$_A$*)...]
EPB
CryptoToken[... ...
(generalID$_B$, sendersID$_B$,random$_B$, time$_B$, Dh$_B$),
({generalID$_B$, sendersID$_B$,random$_B$, time$_B$, Dh$_B$}*Sign*$_B$)...]
Request
ClearToken [...sendersID$_A$, ({generalID$_B$ XOR random$_B$ XOR ...}E$_{DH-secret}$)...]
Phase 2
ClearToken [...generalID$_A$, sendersID$_B$ random$_B$)...]
Response

[... ...] indicates a sequence of tokens.

() indicates a particular token, which may contain multiple elements.

{}E$_{\textbf{DH-secret}}$ indicates the contained values are encrypted utilizing the Diffie-Hellman secret.

EPB knows which shared secret key to use to decipher the **generalID$_B$** identifier by associating it with the **generalID$_A$**, which should also be passed in the message as **sendersID$_A$**. Note that the encrypted value in phase 2 is passed in the **generalID** field of a **clearToken** to simplify encoding.

Figure 4/H.235.0 – Diffie-Hellman with optional authentication

8.2 Subscription-based authentication

Although the procedures outlined here (and the ISO algorithms from which they are derived) are bidirectional in nature, they may be utilized in only one direction if authentication is only needed in that direction. Both two-pass and three-pass procedures are described. The mutual two-pass authentication may be done only in one direction when the messages originating from the reverse direction need not be authenticated. These exchanges assume that each end possesses some well-known identifier (such as a text identifier) which uniquely identifies it. For the two-pass procedure, the further assumption is made that there is a mutually acceptable reference to time (from which to derive timestamps). The amount of time skew that is acceptable is a local implementation matter. The three-pass procedure uses a randomly-generated, unpredictable challenge number (which may be augmented by a sequential counter 'random') as a challenge from the authenticator. This random number is intended to protect against replay attacks. Different to the two-pass procedures, the three-pass procedures do not authenticate the first, initial message holding the initiator's challenge.

There are three different variations that may be implemented depending on requirements:

1) password-based with symmetric encryption;

2) password-based with hashing;

3) certificate-based with signatures.

In all cases, the token will contain the information as described in the following clauses depending on the variation chosen. Note that, in all cases, the **generalID** may be known through configuration or directory lookup rather than in band protocol exchange. To simplify processing at the receiver, the sender should include its identity within **sendersID** and set the **generalID** to the identification of the recipient.

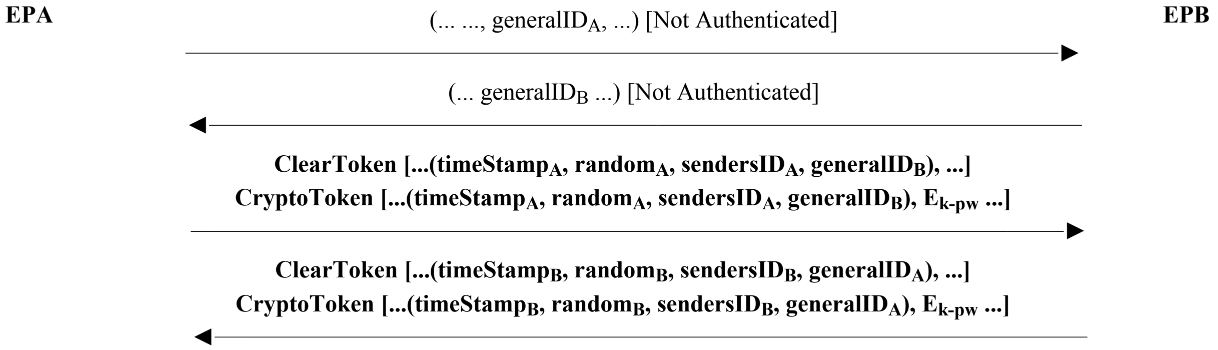
NOTE 1 – In all cases where timestamps are generated and passed as part of a security exchange, implementors should take the following precautions. The timestamp granularity should be fine enough that it is guaranteed to increment with each message. If this is not guaranteed, replay attacks are possible. (e.g., if the timestamp only increments by the minute, then an endpoint "C" can spoof endpoint "A" within the duration of one minute after endpoint "A" has sent a message to endpoint "B").

NOTE 2 – If the message is multicast, then the message is not secured.

8.2.1 Password with symmetric encryption

Figures 5 and 6 show the token format and the message exchange required to perform this type of authentication in two passes or three passes, respectively. This protocol is based on clauses 5.2.1 (two-pass) and 5.2.2 (three-pass) of ISO/IEC 9798-2; it is assumed that an identifier and associated password are exchanged during subscription. The encryption key is length N octets (as indicated by the AlgorithmID), and is formed as follows:

- If password length = N, Key = password;
- if password length < N, the key is padded with zeros;
- if password length > N, the first N octets are assigned to the key, then the N + M*th* octet of the password is XOR'd to the Mmod(N)*th* octet (for all octets beyond N) (i.e., all "extra" password octets are repeatedly folded back on the key by XORing).


EPA
EPB
(... ..., generalID$_A$, ...) [Not Authenticated]
(... generalID$_B$ ...) [Not Authenticated]
ClearToken [...(timeStamp$_A$, random$_A$, sendersID$_A$, generalID$_B$), ...]
CryptoToken [...(timeStamp$_A$, random$_A$, sendersID$_A$, generalID$_B$), E$_{k\text{-}pw}$ ...]
ClearToken [...(timeStamp$_B$, random$_B$, sendersID$_B$, generalID$_A$), ...]
CryptoToken [...(timeStamp$_B$, random$_B$, sendersID$_B$, generalID$_A$), E$_{k\text{-}pw}$ ...]

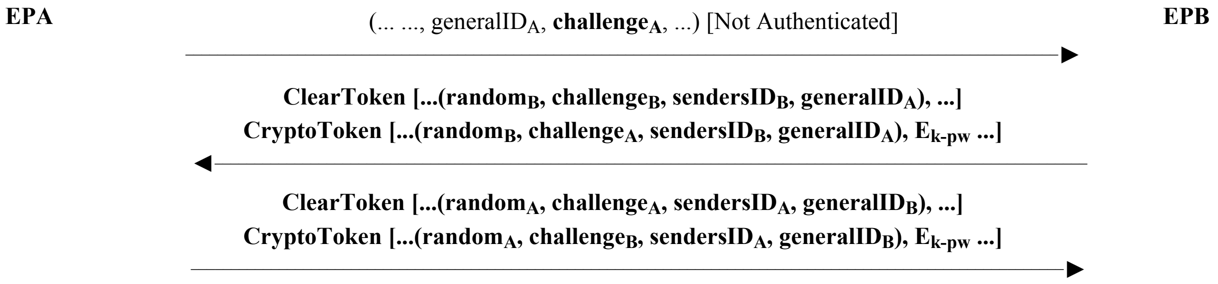
NOTE 1 – The return token from EPB is optional; if omitted, only one-way authentication is achieved.

NOTE 2 – **E$_{k\text{-}pw}$** indicates values that are encrypted using the key "k" derived from the password "pw".

NOTE 3 – **random** is a monotonically increasing counter making multiple messages with the same timestamp unique.

NOTE 4 – In the third message, EPA provides a separate **ClearToken** that is identified through as the same OID as the OID in the **CryptoToken**; similarly for the fourth message and vice versa.

Figure 5/H.235.0 – Password with symmetric encryption; two passes


EPA
EPB
(... ..., generalID$_A$, **challenge$_A$**, ...) [Not Authenticated]
ClearToken [...(random$_B$, challenge$_B$, sendersID$_B$, generalID$_A$), ...]
CryptoToken [...(random$_B$, challenge$_A$, sendersID$_B$, generalID$_A$), E$_{k\text{-}pw}$ ...]
ClearToken [...(random$_A$, challenge$_A$, sendersID$_A$, generalID$_B$), ...]
CryptoToken [...(random$_A$, challenge$_B$, sendersID$_A$, generalID$_B$), E$_{k\text{-}pw}$ ...]

NOTE 1 – **challenge$_A$** and the return encrypted **CryptoToken** from B to A are not necessary if one-way authentication is desired.

NOTE 2 – **E$_{k\text{-}pw}$** indicates an encryption function that is encrypted using the key "k" derived from the password "pw".

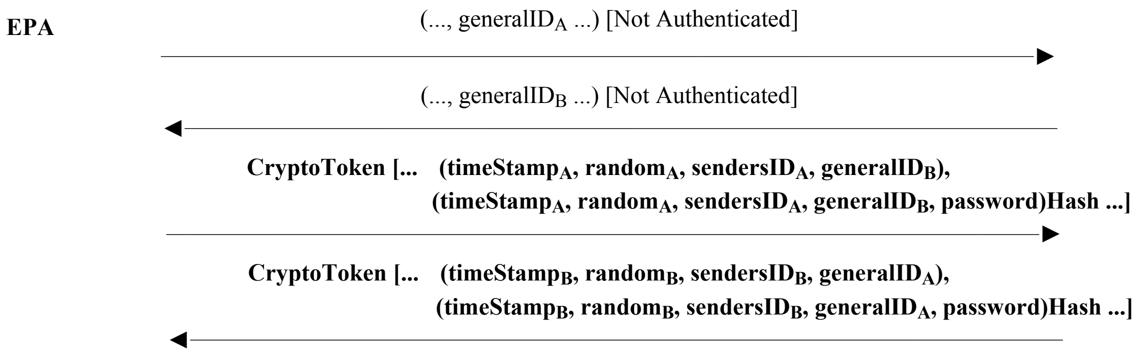
NOTE 3 – In the third message, EPA provides a new **challenge$_A$** in plaintext in a separate **ClearToken**, that is identified through the same OID as the OID in the **CryptoToken**. EPA also returns the encrypted **challenge$_B$** as response; similarly for the second message and vice versa.

NOTE 4 – For multiple outstanding messages, **random** (i.e., a monotonically increasing counter) shall make a challenge unique.

Figure 6/H.235.0 – Password with symmetric encryption; three passes

8.2.2 Password with hashing

Figures 7 and 8 show the token format and the message exchange required to perform this type of authentication for two pass or three passes, respectively. This protocol is based on clauses 5.2.1 and 5.2.2 of ISO/IEC 9798-4; it is assumed that an identifier and associated password are exchanged during subscription. ITU-T Rec. H.235.1 provides detailed description of the two-pass hashing procedure.

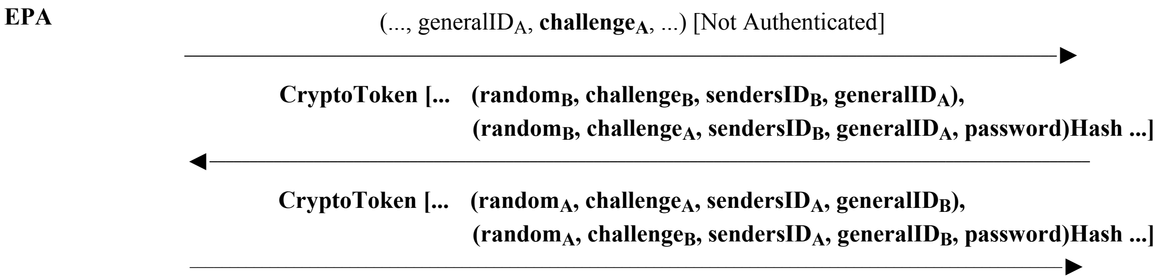

EPA
(..., generalID$_A$ ...) [Not Authenticated]
EPB
(..., generalID$_B$ ...) [Not Authenticated]
CryptoToken [... (timeStamp$_A$, random$_A$, sendersID$_A$, generalID$_B$),
(timeStamp$_A$, random$_A$, sendersID$_A$, generalID$_B$, password)Hash ...]
CryptoToken [... (timeStamp$_B$, random$_B$, sendersID$_B$, generalID$_A$),
(timeStamp$_B$, random$_B$, sendersID$_B$, generalID$_A$, password)Hash ...]

NOTE 1 – The return token from EPB is optional; if omitted, only one-way authentication is achieved.

NOTE 2 – **Hash** indicates a hashing function that operates on the contained values.

NOTE 3 – **random** is a monotonically increasing counter making multiple messages with the same timestamp unique.

Figure 7/H.235.0 – Password with hashing; two passes


EPA
(..., generalID$_A$, **challenge$_A$**, ...) [Not Authenticated]
EPB
CryptoToken [... (random$_B$, challenge$_B$, sendersID$_B$, generalID$_A$),
(random$_B$, challenge$_A$, sendersID$_B$, generalID$_A$, password)Hash ...]
CryptoToken [... (random$_A$, challenge$_A$, sendersID$_A$, generalID$_B$),
(random$_A$, challenge$_B$, sendersID$_A$, generalID$_B$, password)Hash ...]

NOTE 1 – The return token from EPB is optional; if omitted, only one-way authentication is achieved.

NOTE 2 – **Hash** indicates a hashing function that operates on the contained values.

NOTE 3 – In the third message, EPA provides a new **challenge$_A$** in plaintext within the embedded **ClearToken** in **cryptoHashedToken**. EPA also returns the hashed **challenge$_B$** as response; similarly for the second message and vice versa.

NOTE 4 – For multiple outstanding messages, **random** (i.e., a monotonically increasing counter) shall make a challenge unique.

Figure 8/H.235.0 – Password with hashing; three passes

NOTE 1 – The **cryptoHashedToken** structure is used to pass the parameters used in this exchange. Included in this structure are the 'clear' versions of parameters needed to compute the hashed value. Implementors shall include the timestamp in the **hashedVals** and shall *not* include the password. (For example, both the password and the '**generalID**' should be known *a priori* by the recipient; the former may be omitted.)

NOTE 2 – The hashing function shall be applied to the **EncodedGeneralToken** structure that includes at least the ID, timestamp and password fields. The password value shall NOT be passed in the **ClearToken**.

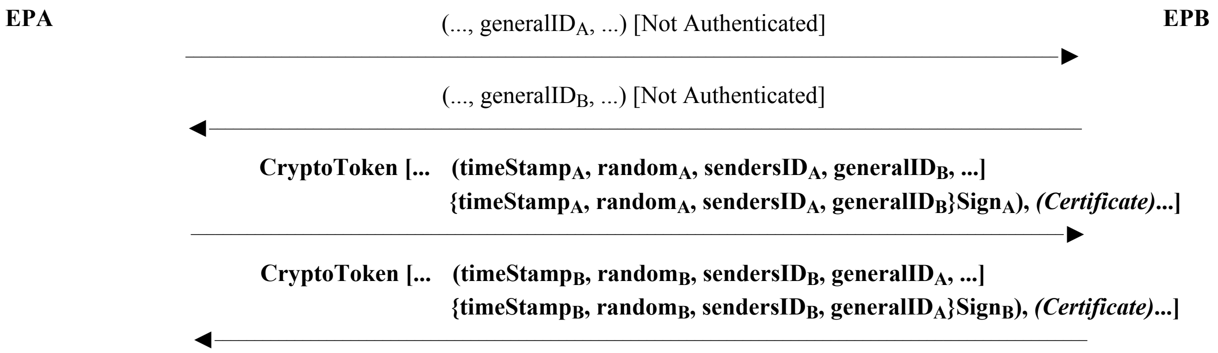
NOTE 3 – Implementations should ensure that user-entered passwords convey sufficient entropy. Passwords that are too short or that are susceptible to dictionary attacks should be rejected. Feeding the user-entered pass-phrase through a cryptographic hash function and using the output bits may be advantageous in certain cases.

8.2.3 Certificate-based with signatures

Figures 9 and 10 show the token format and the message exchange required to perform this type of authentication. This protocol is based on clause 5.2.1 of ISO/IEC 9798-3; it is assumed that an identifier and associated certificate are assigned/exchanged during subscription. ITU-T Rec. H.235.2 provides detailed description of the two-pass signature procedure.

NOTE 1 – An optional certificate element may also be provided; these are illustrated in ***italics*** below.

NOTE 2 – If the message is multicast, then the identifier of the destination (**generalID$_B$** for messages originated at A and vice versa) should not be included in the **ClearToken**.


EPA
EPB
(..., generalID$_A$, ...) [Not Authenticated]
(..., generalID$_B$, ...) [Not Authenticated]
CryptoToken [... (timeStamp$_A$, random$_A$, sendersID$_A$, generalID$_B$, ...]
{timeStamp$_A$, random$_A$, sendersID$_A$, generalID$_B$}Sign$_A$), *(Certificate)*...]
CryptoToken [... (timeStamp$_B$, random$_B$, sendersID$_B$, generalID$_A$, ...]
{timeStamp$_B$, random$_B$, sendersID$_B$, generalID$_A$}Sign$_B$), *(Certificate)*...]

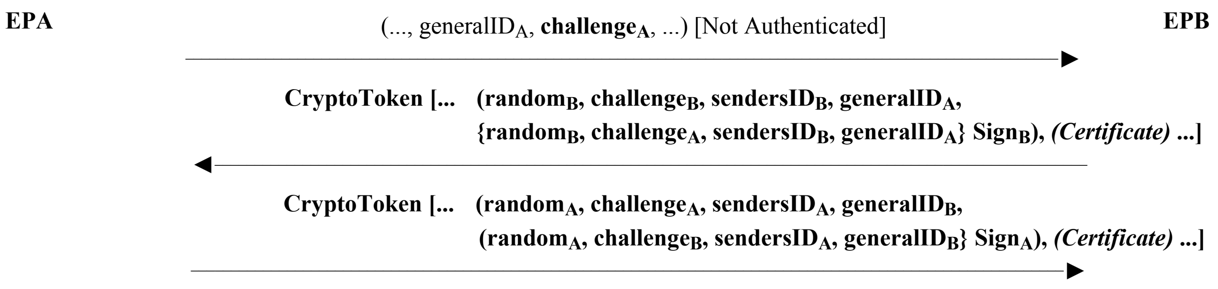
NOTE 1 – The return token from EPB is optional; if omitted, only one-way authentication is achieved.

NOTE 2 – A "payment" type certificate may be optionally included by the EPA originator.

NOTE 3 – **Sign** indicates a signing function (from associated certificate) performed on the contained values.

NOTE 4 – **random** is a monotonically increasing counter making multiple messages with the same timestamp.

Figure 9/H.235.0 – Certificate-based with signatures; two passes


EPA
EPB
(..., generalID$_A$, **challenge$_A$**, ...) [Not Authenticated]
CryptoToken [... (random$_B$, challenge$_B$, sendersID$_B$, generalID$_A$,
{random$_B$, challenge$_A$, sendersID$_B$, generalID$_A$} Sign$_B$), *(Certificate)* ...]
CryptoToken [... (random$_A$, challenge$_A$, sendersID$_A$, generalID$_B$,
(random$_A$, challenge$_B$, sendersID$_A$, generalID$_B$} Sign$_A$), *(Certificate)* ...]

NOTE 1 – The return token from EPB is optional; if omitted, only one-way authentication is achieved.

NOTE 2 – A "payment" type certificate may be optionally included by the EPA originator.

NOTE 3 – **Sign** indicates a signing function (from associated certificate) performed on the contained values.

NOTE 4 – In the third message, EPA provides a new **challenge$_A$** in plaintext within the embedded encoded **GeneralToken**. EPA also returns the signed **challenge$_B$** as response; similarly for the second message and vice versa.

NOTE 5 – For multiple outstanding messages, **random** (i.e., a monotonically increasing counter) shall make a challenge unique.

Figure 10/H.235.0 – Certificate-based with signatures; three passes

8.2.4 Usage of shared secret and passwords

This Recommendation applies certain symmetric cryptographic techniques for the purpose of authentication, integrity and confidentiality. This text uses the term password and shared secret 21 when applying symmetric techniques. Shared secret is understood as the generic term identifying an arbitrary bit string. The shared secret may be assigned or configured as part of the user's subscription process, or may be part of in-band computation such as a Diffie-Hellman-derived shared secret.

A password could be viewed as an alphanumeric character string that users can memorize. It is obvious that using passwords should be done with care. Passwords are able to provide sufficient security only when they are chosen randomly from a large space, when they convey sufficient entropy such that they are unpredictable and when they are changed periodically. Rules for setting up and maintaining passwords do not fall within the scope of this Recommendation.

A good practice as to how to deploy the benefits from passwords and shared secrets is to transform the user password string into a fixed bit string as the shared secret using a cryptographically strong one-way hash function.

As a recommended example, when using the security profile of H.235.1, the SHA1 when applied to the password string, yields to a 20-byte shared secret. An advantage is that the hashed result does not only conceal the actual password, but also defines a fixed length bit string format without really sacrificing entropy.

Thus,

shared secret := SHA1 (password)

8.3 RAS signalling/procedures for authentication

This Recommendation will not explicitly provide any form of message privacy between gatekeepers and endpoints. There are two types of authentication that may be utilized. The first type is symmetric encryption-based that requires no prior contact between the endpoint and gatekeeper. The second type is subscription-based and will have two forms: password or certificate. All of these forms are derived from the procedures shown in clauses 8, 8.2.1, 8.2.2 and 8.2.3. In this Recommendation, the generic labels (EPA and EPB) shown in the aforementioned clauses will represent the endpoint and gatekeeper respectively.

8.3.1 Endpoint-gatekeeper authentication (non-subscription-based)

This mechanism may provide the gatekeeper with a cryptographic link. The cryptographic link asserts that a particular endpoint which previously was registered, is the same one that issues subsequent RAS messages. It should be noted that this may not provide any authentication of the gatekeeper to the endpoint, unless the optional signature element is included. The establishment of the identity relationship occurs when the terminal issues the **GRQ**, as outlined in 7.2.1/H.323. The Diffie-Hellman exchange shall occur in conjunction with the **GRQ** and **GCF** messages as shown in the first phase of clause 8. This shared secret key shall now be used on any subsequent **RRQ**/**URQ** from the terminal to the gatekeeper. If a gatekeeper operates in this mode and receives a **GRQ** without a token containing the *DHset* or an acceptable algorithm value, it shall return a **securityDenial** reason code or other appropriate security error code according to 11.1 in the **DRJ**.

The Diffie-Hellman shared secret key as created during the **GRQ**/**GCF** exchange may be used for authentication on subsequent **xRQ** messages. The following procedures shall be used to complete this mode of authentication.

Terminal **(xRQ)**

1) The terminal shall provide all of the information in the message as described in the appropriate clauses of ITU-T Rec. H.225.0.

2) The terminal shall encrypt the **GatekeeperIdentifier** (as returned in the **GCF**) using the shared secret key that was negotiated. This shall be passed in a **clearToken** (see 8.1) as the **generalID**.

The 16 bits of the **random** and then the **requestSeqNum** shall be XOR'd with each 16 bits of the **GatekeeperIdentifier**. If the **GatekeeperIdentifier** does not end on an even 16 boundary, the last

8 bits of the **GatekeeperIdentifier** shall be XOR'd with the least significant octet of the random value and then **requestSeqNum**. The **GatekeeperIdentifier** shall be encrypted using the selected algorithm in the **GCF** (algorithmOID) and utilizing the entire shared secret.

The following example illustrates this procedure:

RND16: 16-bit value of the Random Value

SQN16: 16-bit value of requestSeqNum

BMPX: the Xth BMP character of GatekeeperIdentifier

BMP1' = (BMP1) XOR (RND16) XOR (SQN16)

BMP2' = (BMP2) XOR (RND16) XOR (SQN16)

BMP3' = (BMP3) XOR (RND16) XOR (SQN16)

BMP4' = (BMP4) XOR (RND16) XOR (SQN16)

BMP5' = (BMP5) XOR (RND16) XOR (SQN16)

:

:

BMPn' = (BMPn) XOR (RND16) XOR (SQN16)

In order to cryptographically link this and subsequent messages with the original registrant (the endpoint that issued the **RRQ**), the most recent **random** value returned shall be utilized (this value may be one newer than the value returned in the **RCF** from a later **xCF** message).

Gatekeeper **(xCF/xRJ)**

1) Gatekeeper shall encrypt its **GatekeeperIdentifier** (following the above procedure) with the shared secret key associated with the endpoint alias and compare this to the value in the **xRQ**.
2) Gatekeeper shall return **xRJ** if the two encrypted values do not match.
3) If **GatekeeperIdentifier** matches, gatekeeper shall apply any local logic and respond with **xCF** or **xRJ**.
4) If an **xCF** is sent by the gatekeeper, it should contain an assigned **EndpointIdentifier** and a new random value in the **random** field of a **clearToken**.

Refer to the second phase of Figure 4 for a graphical representation of this exchange. The gatekeeper knows which shared secret key to use to decipher the gatekeeper identifier by the alias name in the message.

8.3.2 Endpoint-gatekeeper authentication (subscription-based)

All RAS messages other than **GRQ**/**GCF** should contain the authentication tokens required by the specific mode of operation. There are three different variations that may be implemented depending on requirements and environment:

1) password-based with symmetric encryption;
2) password-based with hashing;
3) certificate-based with signatures.

In all cases, the token will contain the information as described in the following clauses depending on the variation chosen. If a gatekeeper operates in a secure mode and receives an RAS message without an acceptable token value, it shall return a **securityDenial** reason code or other appropriate security error code according to 11.1 in the reject message. In all cases, the return token from GK is optional; if omitted, only one-way authentication is achieved.

8.3.2.1 Password with symmetric encryption

The gatekeeper discovery phase (**GRQ**, **GCF** and **GRJ**) may be unsecured as shown in Figure 11, or may be secured using the **cryptoTokens**.

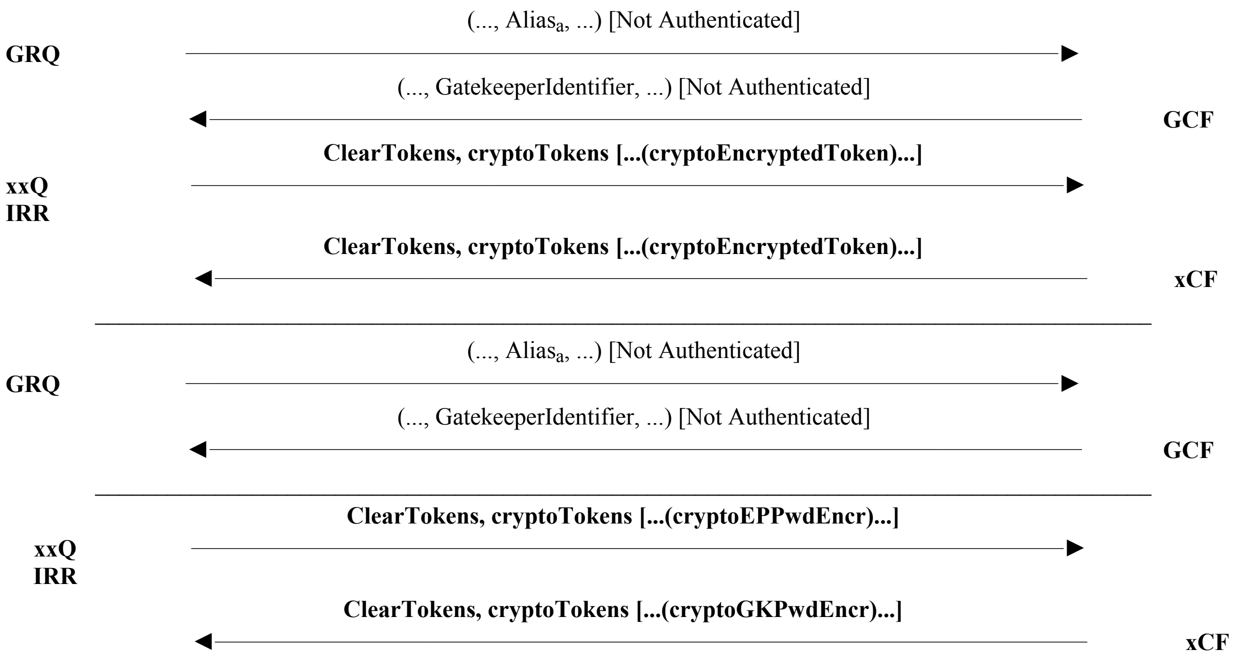

(..., Alias$_a$, ...) [Not Authenticated]
GRQ
(..., GatekeeperIdentifier, ...) [Not Authenticated]
GCF
ClearTokens, cryptoTokens [...(cryptoEncryptedToken)...]
xxQ
IRR
ClearTokens, cryptoTokens [...(cryptoEncryptedToken)...]
xCF
(..., Alias$_a$, ...) [Not Authenticated]
GRQ
(..., GatekeeperIdentifier, ...) [Not Authenticated]
GCF
ClearTokens, cryptoTokens [...(cryptoEPPwdEncr)...]
xxQ
IRR
ClearTokens, cryptoTokens [...(cryptoGKPwdEncr)...]
xCF

Figure 11/H.235.0 – Password with symmetric encryption

8.3.2.2 Password with hashing

The gatekeeper discovery phase (**GRQ**, **GCF** and **GRJ**) may be unsecured as shown in Figure 12, or may be secured according to ITU-T Rec. H.235.1 using the **cryptoTokens**.

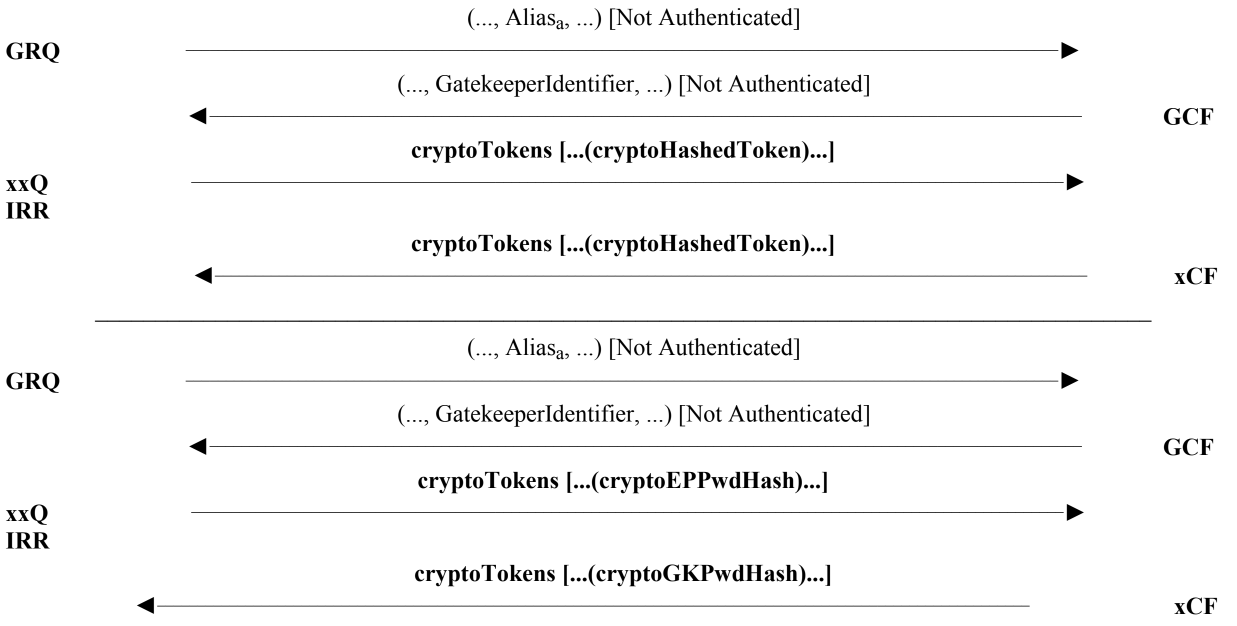

(..., Alias$_a$, ...) [Not Authenticated]
GRQ
(..., GatekeeperIdentifier, ...) [Not Authenticated]
GCF
cryptoTokens [...(cryptoHashedToken)...]
xxQ
IRR
cryptoTokens [...(cryptoHashedToken)...]
xCF
(..., Alias$_a$, ...) [Not Authenticated]
GRQ
(..., GatekeeperIdentifier, ...) [Not Authenticated]
GCF
cryptoTokens [...(cryptoEPPwdHash)...]
xxQ
IRR
cryptoTokens [...(cryptoGKPwdHash)...]
xCF

Figure 12/H.235.0 – Password with hashing

8.3.2.3 Certificate-based with signatures

The gatekeeper discovery phase (**GRQ**, **GCF** and **GRJ**) may be unsecured as shown in Figure 13, or may be secured according to ITU-T Rec. H.235.2 using the **cryptoTokens**.

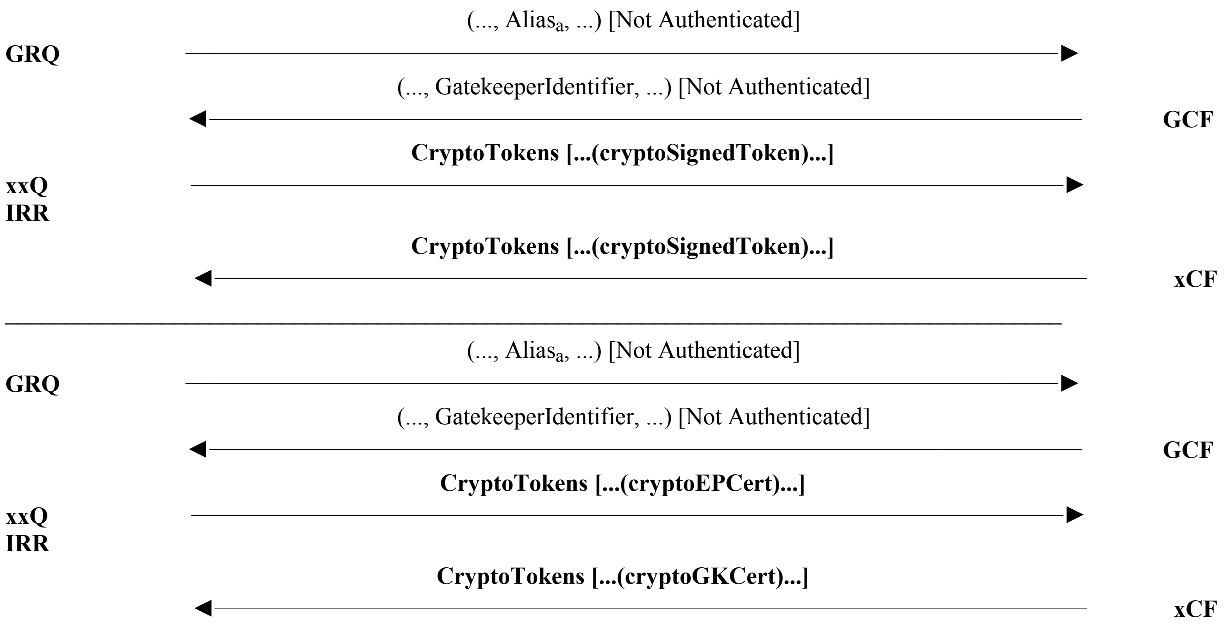

(..., Alias$_a$, ...) [Not Authenticated]
GRQ
(..., GatekeeperIdentifier, ...) [Not Authenticated]
GCF
CryptoTokens [...(cryptoSignedToken)...]
xxQ
IRR
CryptoTokens [...(cryptoSignedToken)...]
xCF
(..., Alias$_a$, ...) [Not Authenticated]
GRQ
(..., GatekeeperIdentifier, ...) [Not Authenticated]
GCF
CryptoTokens [...(cryptoEPCert)...]
xxQ
IRR
CryptoTokens [...(cryptoGKCert)...]
xCF

Figure 13/H.235.0 – Certificate-based with signatures

8.4 Key management on the RAS channel

In some circumstances, it is desirable to distribute (RAS) session keys from a gatekeeper to one or more endpoints under its control, or from one endpoint to another. The proposed mechanism assumes that the gatekeeper and the endpoint share a strong, secret key or know each other's public key. One example of such a case would be for a routing gatekeeper to issue a session key to an endpoint in a RAS message, such as **RCF** or **ACF**, for use in encrypting a gatekeeper-routed signalling channel. Another example might be one in which the gatekeeper issues a session key for use in encrypting succeeding RAS communications (e.g., **RRQ** or **ARQ**).

This mechanism is similar to that used for distribution of media session keys. It may be used to avoid the overhead of key negotiation in certain circumstances.

For key transport, the optional **h235Key** field of the **ClearToken** should be used in H.235v3 or higher. The flexibility of the **H235Key** element will permit the transport of encryption key material using:

- a secure channel (the **secureChannel** option) assuming the RAS or call signalling channel is secured by other means (IPsec/SSL, etc.);
- a shared encryption secret over a clear channel (the **sharedSecret** choice), or similarly but preferably the **secureSharedSecret** choice;
- a public-key encryption and certificate over a clear channel (the **certProtectedKey** option).

The usage of the exchanged RAS session key and its application to RAS, call signalling messages and/or transport channels is left as for further study.

9 Asymmetric authentication and key exchange using elliptic curve crypto systems

This Recommendation provides sophisticated elliptic curve techniques with applications to signature, key management and encryption. One of the primary advantages over "classical" asymmetric techniques such as RSA are:

- Shorter cryptographic keys yielding comparable security as RSA: Typical key lengths for elliptic curve crypto systems are 160 bits; i.e., equivalent in security to a 1024-bit RSA key. The shorter key consumes less memory for storage and makes elliptic curve crypto systems especially attractive for implementation in smart-cards, and in any other devices with low memory requirements. In the H.323 environment, Annex J/H.323-based secured audio simple endpoint types (SASETs) with their low price requirements are well-suited for deployment of elliptic curve techniques.
- Improved processing speed achieved both in software and in hardware implementations: The shorter keys contribute to the processing speed. This results in faster interactive (user) responses.

All the background information, explanation and processing procedures of elliptic curve cryptography can be found in *ATM Security Specification Version 1.1*, section 8.7. It is recommended to encode the elliptic points in their affine, uncompressed notation without using the point-compression/decompression method. Further information on this topic is available in ISO/IEC 15946-1 and ISO/IEC 15946-2.

9.1 Key management

Elliptic curve-based Diffie-Hellman key agreement schemes are similar to the classic mod-p case as defined in this Recommendation as well. There are two cases:

- elliptic curves over a prime field: **eckasdhp** holds the elliptic curve and Diffie-Hellman parameters;
- elliptic curves of characteristic 2: **eckasdh2** holds the elliptic curve and Diffie-Hellman parameters.

The ECKASDH structure holds either case. Some example elliptic curves are listed in ISO/IEC 15946-1. Any other suitable and appropriate elliptic curves could be used as well.

Due to the available sequenced structure of the **ClearToken** signalling, both **dhkey** and **eckasdhkey** should not occur at the same time; only one shall be present when the Diffie-Hellman key exchange is applied.

Remark – Do not confuse the randomly chosen secret parameters ***a*** by party A or ***b*** by party B with the common Weierstrass coefficients ***a, b***.

9.2 Digital signature

The **ECGDSASignature** field carries the values **r** and **s** of the computed elliptic curved-based digital signature. Section 8.7.3 of *ATM Security Specification Version 1.1* and chapter 5 of ISO/IEC 15946-2 provide further information on the signature algorithm EC-GDSA.

The elliptic curve-based digital signature **ECGDSA** shall be ASN.1 coded and then put into the **signature** field of the **SIGNED** macro of this Recommendation. For the digital signature, the sender shall include an object identifier into **algorithmOID** by which the recipient is able to determine usage of an elliptic curve digital signature.

10 Pseudo-Random Function (PRF)

This clause defines a pseudo-random function for the purpose of deriving dynamic keys from a static key material and a random value.

NOTE – This PRF is identical to the MIKEY PRF (see RFC 3830 section 4.1.2).

The key derivation method has the following input parameters:

- *inkey*: the input key to the derivation function.
- *inkey_len*: the length in bits of the input key.
- *label*: a specific label, dependent on the type of the key to be derived and the random **challenge** value.
- *outkey_len*: desired length in bits of the output key.

The pseudo-random function has the following output:

- *outkey*: the output key of desired length.

This PRF shall use the PRF as is defined in RFC 3830 section 4.1.2.

11 Security error recovery

This Recommendation does not specify or recommend any methods by which endpoints may monitor their absolute privacy. It does, however, recommend actions to be taken when privacy loss is detected.

If either endpoint detects a breach in the security of the call connection channel (e.g., H.225.0 for H.323), it should immediately close the connection following the protocol procedures appropriate to the particular endpoint (for 8.5/H.323 with the exception of step B-5).

If either endpoint detects a breach in the security of the H.245 channel or the secured data (**h235Control**) logical channel, it should immediately close the connection following the protocol procedures appropriate to the particular endpoint (for 8.5/H.323 with the exception of step B-5).

If any endpoint detects a loss of privacy on one of the logical channels, it should immediately request a new key **(encryptionUpdateRequest)** and/or close the logical channel. At the discretion of the MC(U), a loss of privacy on one logical channel may cause all other logical channels to be closed and/or re-keyed at the discretion of the MC(U). MC(U) shall forward **encryptionUpdateRequest, encryptionUpdate** to any and all endpoints affected.

At the discretion of the MC(U), a security error on an individual channel may cause the connections to be closed on all of the conference endpoints, thus ending the conference.

11.1 Error signalling

A security capable gatekeeper or other security enhanced H.225.0 entity shall provide error indications. The security error indicates that the entity was not able to correctly process the received message. Whenever possible, a detailed error code shall be provided.

- **securityWrongSyncTime** shall indicate that the sender found a security problem with inappropriate timestamps. This could be caused due to a problem with the time server, lost synchronization or due to excessive network delay.
- **securityReplay** shall indicate that a replay attack has been encountered. This is the case when the same sequence number occurs more than once for a given timestamp.
- **securityWrongGeneralID** shall indicate a mismatch of the general ID in the message. This could be caused due to wrong addressing.
- **securityWrongSendersID** shall indicate a mismatch of the sender's ID in the message. This could be caused due to user's erroneous entry.
- **securityIntegrityFailed** shall indicate that the integrity/signature check failed. For H.235.1, this could be caused due to a wrong or mistyped password during the initial request or due to an encountered active attack. For H.235.2 and H.235.3, this shall indicate

that the digital signature check upon the message failed. This could be caused due to a wrong private/public key applied or due to an encountered active attack.

- **securityWrongOID** shall indicate any mismatch in token OIDs (clear or crypto token) or crypto algorithm OIDs. This indicates different security algorithms/profiles implemented.
- **securityDHmismatch** shall indicate any mismatch in the Diffie-Hellman parameters exchanged. This might indicate different DH-parameter sets or even different voice encryption algorithms implemented.
- **securityCertificateExpired** shall indicate that a certificate has expired.
- **securityCertificateDateInvalid** shall indicate that a certificate is not yet valid.
- **securityCertificateRevoked** shall indicate that a certificate was found revoked.
- **securityCertificateNotReadable** shall indicate that a certificate could not be correctly ASN.1 decoded or is in other bad shape.
- **securityCertificateSignatureInvalid** shall indicate that the signature in the certificate is not correct.
- **securityCertificateMissing** shall indicate that a certificate was expected but found missing or that the certificate could not be located otherwise.
- **securityCertificateIncomplete** shall indicate that some expected certificate extensions were not present.
- **securityUnsupportedCertificateAlgOID** shall indicate that certain crypto algorithms such as hash or digital signatures used within the certificate are not understood or are not supported. As part of the returned response, the sender may provide a list of acceptable certificates in separate tokens in order to facilitate selection of an appropriate one by the recipient.
- **securityUnknownCA** shall indicate that the CA/root certificate could not be found or that the certificate could not be matched with a trusted CA.

In any other case where the H.235 security operation has failed, **securityDenial** for H.225.0 RAS (**securityDenied** for H.225.0 call signalling resp.) shall be returned.

NOTE 1 – securityWrongSyncTime, securityReplay, securityWrongGeneralID, securityWrongSendersID, SecurityIntegrityFailed, securityDHmismatch, and securityWrongOID may occur in H.235.1, H.235.2 or in H.235.3 security profiles.

NOTE 2 – securityCertificateExpired, securityCertificateDateInvalid, securityCertificateRevoked, securityCertificateNotReadable, securityCertificateSignatureInvalid, securityCertificateMissing, securityCertificateIncomplete, securityUnsupportedCertificateAlgOID and securityUnknownCA may occur in H.235.2 or in H.235.3 security profiles.

Annex A

H.235 ASN.1

```
H235-SECURITY-MESSAGES DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

-- EXPORTS All

ChallengeString        ::= OCTET STRING (SIZE(8..128))
TimeStamp              ::= INTEGER(1..4294967295)    -- seconds since 00:00
                                                     -- 1/1/1970 UTC
RandomVal              ::= INTEGER -- 32-bit Integer
Password               ::= BMPString (SIZE (1..128))
Identifier             ::= BMPString (SIZE (1..128))
KeyMaterial            ::= BIT STRING(SIZE(1..2048))

NonStandardParameter ::= SEQUENCE
{
     nonStandardIdentifier   OBJECT IDENTIFIER,
     data                    OCTET STRING
}

-- if local octet representations of these bit strings are used they shall
-- utilize standard Network Octet ordering (e.g., Big Endian)
DHset ::= SEQUENCE
{
     halfkey        BIT STRING (SIZE(0..2048)), -- = g^x mod n
     modSize        BIT STRING (SIZE(0..2048)), --  n
     generator      BIT STRING (SIZE(0..2048)), -- g
     ...
}

ECpoint ::= SEQUENCE -- uncompressed (x, y) affine coordinate representation of
                     -- an elliptic curve point
{
     x         BIT STRING (SIZE(0..511)) OPTIONAL,
     y         BIT STRING (SIZE(0..511)) OPTIONAL,
     ...
}

ECKASDH::= CHOICE -- parameters for elliptic curve key agreement scheme Diffie-
Hellman
{
     eckasdhp SEQUENCE -- parameters for elliptic curves of prime field
     {
          public-key     ECpoint, -- This field contains representation of
               -- the ECKAS-DHp public key value. This field contains the
               -- initiator's ECKAS-DHp public key value (aP) when this
               -- information element is sent from originator to receiver. This
               -- field contains the responder's ECKAS-DHp public key value (bP)
               -- when this information element is sent back from receiver to
               -- originator.
          modulus        BIT STRING (SIZE(0..511)), -- This field contains
               -- representation of the ECKAS-DHp public modulus value (p).
          base           ECpoint, -- This field contains representation of the
               -- ECKAS-DHp public base (P).
          weierstrassA   BIT STRING (SIZE(0..511)), -- This field contains
               -- representation of the ECKAS-DHp Weierstrass coefficient (a).
          weierstrassB   BIT STRING (SIZE(0..511)) -- This field contains
               -- representation of the ECKAS-DHp Weierstrass coefficient (b).
     },
     eckasdh2 SEQUENCE -- parameters for elliptic curves of characteristic 2
     {
         public-key    ECpoint, -- This field contains representation of
              -- the ECKAS-DH2 public key value.
              -- This field contains the initiator's ECKAS-DH2 public key value
              -- (aP) when this information element is sent from originator to
              -- receiver. This field contains the responder's ECKAS-DH2 public
              -- key value (bP) when this information element is sent back from
              -- receiver to originator.
         fieldSize     BIT STRING (SIZE(0..511)), -- This field contains
              -- representation of the ECKAS-DH2 field size value (m).
         base          ECpoint, -- This field contains representation of the
              -- ECKAS-DH2 public base (P).
         weierstrassA  BIT STRING (SIZE(0..511)), -- This field contains
              -- representation of the ECKAS-DH2 Weierstrass coefficient (a).
         weierstrassB  BIT STRING (SIZE(0..511)) -- This field contains
              -- representation of the ECKAS-DH2 Weierstrass coefficient (b).
     },
     ...
}

ECGDSASignature::= SEQUENCE -- parameters for elliptic curve digital signature
              -- algorithm
{
     r        BIT STRING (SIZE(0..511)), -- This field contains the
              -- representation of the r component of the ECGDSA digital
              -- signature.
     s        BIT STRING (SIZE(0..511)) -- This field contains the
              -- representation of the s component of the ECGDSA digital
              -- signature.
}

TypedCertificate ::= SEQUENCE
{
     type          OBJECT IDENTIFIER,
     certificate   OCTET STRING,
     ...
}

AuthenticationBES ::= CHOICE

{
     default       NULL, -- encrypted ClearToken
     radius        NULL, -- RADIUS-challenge/response
     ...

}

AuthenticationMechanism  ::= CHOICE
{
     dhExch        NULL, -- Diffie-Hellman
     pwdSymEnc     NULL, -- password with symmetric encryption
     pwdHash       NULL, -- password with hashing
     certSign      NULL, -- Certificate with signature
     ipsec         NULL, -- IPSEC based connection
     tls           NULL,
     nonStandard   NonStandardParameter, -- something else.
     ...,
     authenticationBES  AuthenticationBES, -- user authentication for BES
     keyExch  OBJECT IDENTIFIER -- key exchange profile
}
```

```
ClearToken          ::= SEQUENCE  -- a "token" may contain multiple value types.
{
     tokenOID       OBJECT IDENTIFIER,
     timeStamp      TimeStamp OPTIONAL,
     password       Password OPTIONAL,
     dhkey          DHset OPTIONAL,
     challenge      ChallengeString OPTIONAL,
     random         RandomVal OPTIONAL,
     certificate    TypedCertificate OPTIONAL,
     generalID      Identifier OPTIONAL,
     nonStandard    NonStandardParameter OPTIONAL,
     ...,
     eckasdhkey     ECKASDH OPTIONAL,  -- elliptic curve Key Agreement
                                       -- Scheme-Diffie Hellman Analogue
                                       -- (ECKAS-DH)
     sendersID      Identifier OPTIONAL,
     h235Key        H235Key OPTIONAL, -- central distributed key in V3
     profileInfo    SEQUENCE OF ProfileElement OPTIONAL  -- profile-specific
}

--   An object identifier should be placed in the tokenOID field when a
--   ClearToken is included directly in a message (as opposed to being
--   encrypted). In all other cases, an application should use the
--   object identifier { 0 0 } to indicate that the tokenOID value is not
--   present.
--   Start all the cryptographic parameterized types here...
--

ProfileElement      ::= SEQUENCE
{
     elementID      INTEGER (0..255), -- element identifier, as defined by
                                      -- profile
     paramS         Params OPTIONAL,  -- any element-specific parameters
     element        Element OPTIONAL, -- value in required form
     …
}

Element ::= CHOICE
{
     octets              OCTET STRING,
     integer             INTEGER,
     bits                BIT STRING,
     name                BMPString,
     flag                BOOLEAN,
     …
}



SIGNED { ToBeSigned } ::= SEQUENCE {
     toBeSigned          ToBeSigned,
     algorithmOID        OBJECT IDENTIFIER,
     paramS              Params,   -- any "runtime" parameters
     signature           BIT STRING -- could be an RSA or an ASN.1 coded
ECGDSA Signature
} ( CONSTRAINED BY { -- Verify or Sign Certificate -- } )


ENCRYPTED { ToBeEncrypted } ::= SEQUENCE {
     algorithmOID        OBJECT IDENTIFIER,
     paramS              Params,   -- any "runtime" parameters
     encryptedData       OCTET STRING
} ( CONSTRAINED BY { -- Encrypt or Decrypt -- ToBeEncrypted } )
```

```
HASHED { ToBeHashed } ::= SEQUENCE {
     algorithmOID       OBJECT IDENTIFIER,
     paramS             Params,   -- any "runtime" parameters
     hash               BIT STRING
} ( CONSTRAINED BY { -- Hash -- ToBeHashed } )

IV8 ::= OCTET STRING (SIZE(8)) -- initial value for 64-bit block ciphers
IV16 ::= OCTET STRING (SIZE(16)) -- initial value for 128-bit block ciphers

-- signing algorithm used must select one of these types of parameters
-- needed by receiving end of signature.

Params ::= SEQUENCE {
     ranInt         INTEGER OPTIONAL, -- some integer value
     iv8            IV8 OPTIONAL, -- 8-octet initialization vector
     ...,
     iv16           IV16 OPTIONAL,-- 16-octet initialization vector
     iv             OCTET STRING OPTIONAL, -- arbitrary length initialization
     vector
     clearSalt      OCTET STRING OPTIONAL -- unencrypted salting key for
     encryption
}

EncodedGeneralToken ::= TYPE-IDENTIFIER.&Type (ClearToken -- general usage token
-- )
PwdCertToken ::= ClearToken (WITH COMPONENTS {..., timeStamp PRESENT, generalID
PRESENT})
EncodedPwdCertToken ::= TYPE-IDENTIFIER.&Type (PwdCertToken)

CryptoToken::= CHOICE
{

     cryptoEncryptedToken SEQUENCE -- General purpose/application specific token
     {
          tokenOID      OBJECT IDENTIFIER,
          token         ENCRYPTED { EncodedGeneralToken }
     },
     cryptoSignedToken  SEQUENCE -- General purpose/application specific token
     {
          tokenOID      OBJECT IDENTIFIER,
          token         SIGNED { EncodedGeneralToken }
     },
     cryptoHashedToken SEQUENCE -- General purpose/application specific token
     {
          tokenOID           OBJECT IDENTIFIER,
          hashedVals         ClearToken,
          token HASHED { EncodedGeneralToken }
     },
     cryptoPwdEncr ENCRYPTED { EncodedPwdCertToken },
     ...
}

-- These allow the passing of session keys within the H.245 OLC structure.
-- They are encoded as standalone ASN.1 and based as an OCTET STRING within
-- H.245
H235Key  ::=CHOICE  -- This is used with the H.245 or ClearToken "h235Key"
field
{
     secureChannel           KeyMaterial,
     sharedSecret            ENCRYPTED {EncodedKeySyncMaterial},
     certProtectedKey        SIGNED {EncodedKeySignedMaterial },
     ...,
     secureSharedSecret      V3KeySyncMaterial -- for H.235 V3 endpoints
}
```

```
KeySignedMaterial ::= SEQUENCE {
     generalId     Identifier, -- slave's alias
     mrandom       RandomVal, -- master's random value
     srandom       RandomVal OPTIONAL, -- slave's random value
     timeStamp     TimeStamp OPTIONAL, -- master's timestamp for unsolicited EU
     encrptval     ENCRYPTED { EncodedKeySyncMaterial }
}
EncodedKeySignedMaterial ::= TYPE-IDENTIFIER.&Type (KeySignedMaterial)

H235CertificateSignature    ::= SEQUENCE
{
     certificate            TypedCertificate,
     responseRandom         RandomVal,
     requesterRandom        RandomVal OPTIONAL,
     signature              SIGNED { EncodedReturnSig },
     ...
}

ReturnSig ::= SEQUENCE {
     generalId              Identifier, -- slave's alias
     responseRandom         RandomVal,
     requestRandom          RandomVal OPTIONAL,
     certificate            TypedCertificate OPTIONAL -- requested certificate
}

EncodedReturnSig ::= TYPE-IDENTIFIER.&Type (ReturnSig)
KeySyncMaterial    ::= SEQUENCE
{
     generalID     Identifier,
     keyMaterial   KeyMaterial,
     ...
}
EncodedKeySyncMaterial  ::=TYPE-IDENTIFIER.&Type (KeySyncMaterial)



V3KeySyncMaterial  ::= SEQUENCE
{
     generalID              Identifier OPTIONAL, -- peer terminal ID
     algorithmOID           OBJECT IDENTIFIER OPTIONAL, -- encryption algorithm
     paramS                 Params, -- IV
     encryptedSessionKey    OCTET STRING OPTIONAL, -- encrypted session key
     encryptedSaltingKey    OCTET STRING OPTIONAL, -- encrypted media salting
                                                   -- key
     clearSaltingKey        OCTET STRING OPTIONAL, -- unencrypted media salting
                                                   -- key
     paramSsalt             Params OPTIONAL, -- IV (and clear salt) for salting
                                                   -- key encryption
     keyDerivationOID       OBJECT IDENTIFIER OPTIONAL, -- key derivation
                                                   -- method
     ...,
     genericKeyMaterial     OCTET STRING OPTIONAL -- ASN.1-encoded key material
                        -- form is dependent on associated media encryption tag

}


END  -- End of H235-SECURITY-MESSAGES DEFINITIONS
```

Annex B

H.324-specific topics

For further study.

Appendix I

H.323 implementation details

I.1 Implementation examples

The following subclauses describe example implementations that might be developed within the H.235 framework. These are not intended to constrain the many other possibilities available within this Recommendation, but rather to give more concrete examples of usage within ITU-T Rec. H.323.

I.1.1 Tokens

This clause will describe an example usage of security tokens to obscure or hide destination addressing information. The example scenario is an endpoint which wishes to make a call to another endpoint utilizing its well-known alias. More specifically, this involves an H.323 endpoint, gatekeeper, POTS-gateway, and telephone as illustrated in Figure I.1.

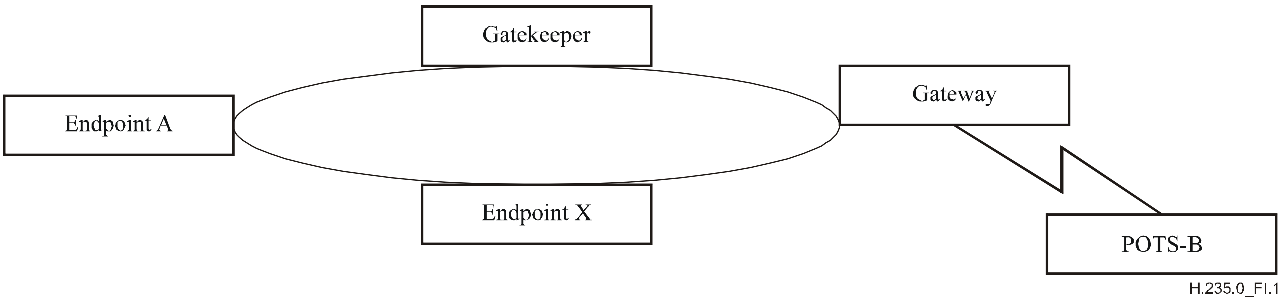


Figure I.1/H.235.0 – Tokens

Currently, H.323 may operate in a manner similar to a telephone network with caller-ID. This scenario will illustrate a situation in which the *caller* does not want to expose its physical address, while still allowing the call to complete. This may be important in POTS-H.323 gateways, where the target phone number may need to stay private.

Assume that EPA is trying to call POTS-B, and POTS-B does not want to expose its E.164 phone number to EPA. (How this policy is established is beyond the scope of this example.)

- EPA will send an **ARQ** to its gatekeeper to resolve the address of the POTS telephone as represented by its alias/GW. The gatekeeper would recognize this as a "private" alias, knowing that in order to complete the connection it must return the POTS-gateway address (similar to returning the address of an H.320 gateway if an H.320 endpoint is called by an H.323 endpoint).
- In the returned **ACF**, the gatekeeper returns the POTS-gateway's address as expected. The addressing information that is required to dial to the end telephone (i.e., the telephone number) is returned in an encrypted token included in the **ACF**. This encrypted token

contains the actual E.164 (phone number) of the telephone which cannot be deciphered nor understood by the caller (i.e., EPA).

- The endpoint issues the SETUP message to the gateway device (whose call signalling address was returned in the **ACF**) including the opaque token(s) that it received with the **ACF**.
- The gateway, upon receiving the SETUP, issues its **ARQ** to its gatekeeper, including any token(s) that were received in the SETUP.
- The gatekeeper is able to decipher the token(s) and return the phone number in the **ACF**.

Partial ASN.1 of an example token structure is shown below, with the field contents described. Assume we utilize the **cryptoEncodedGeneralToken** to contain the encrypted telephone number.

An implementation might choose a **tokenOID** denoting this token as containing the E.164 phone number. The particular method that is used to encrypt this phone number (for example, 56-bit DES) would be included in the "ENCRYPT" definition **algorithmOID**.

```
CryptoToken::= CHOICE
{
    cryptoEncodedGeneralToken SEQUENCE   -- General purpose/application
                                         --  specific token
    {
        tokenOID  OBJECT IDENTIFIER,
        ENCRYPTED { EncodedGeneralToken }
    },
.
.
. [abbreviated text]
.

}
```

The **CryptoToken** would be passed in the SETUP (from EPA to GW) and the **ARQ** (from the GW to the gatekeeper) messages as outlined above. After the gatekeeper decrypted the token (the telephone number) it would pass the clear version of this in the **clearToken**.

I.1.2 Token usage in H.323 systems

There has been some confusion on the usage of individual **CryptoH323Tokens** as passed in RAS messages. There are two main categories of **CryptoH323Tokens**: those used for H.235 procedures and those used in an application-specific manner. The use of these tokens should be according to the following rules:

- All H.235-defined (e.g., **cryptoEPPwdHash**, **cryptoGKPwdHash**, **cryptoEPPwdEncr**, **cryptoGKPwdEncr**, **cryptoGKCert**, and **cryptoFastStart**) shall be utilized with the procedures and algorithms as described in this Recommendation.
- Application-specific or proprietary use of tokens shall utilize the **nestedcryptoToken** for their exchanges.
- Any **nestedcryptoToken** used should have a **tokenOID** (object identifier) which unambiguously identifies it.

I.1.3 H.235 random value usage in H.323 systems

The random value that is passed in **xRQ**/**xCF** sequence between endpoints and gatekeepers may be updated by the gatekeeper. As described in 8.3.1, this random value may be refreshed in any **xCF** message to be utilized by a subsequent **xRQ** messages from the endpoint. Due to the fact that RAS messages may be lost (including **xCF**/**xRJ**), the updated random value may also be lost. The recovery from this situation may be the reinitializing of the security context but is left to local implementation.

Implementations that require the use of multiple outstanding RAS requests will be limited by the updating of the random values used in any authentication. If the updating of this value occurs on every response to a request, parallel requests are not possible. One possible solution is to have a logical "window" during which a random value remains constant. This issue is a local implementation matter.

I.1.4 Password

In this example, it is assumed that the user is a subscriber to the gatekeeper (i.e., the user will be in its zone) and has an associated subscription ID and password. The user would register with the gatekeeper using the subscription ID (as passed in an alias – H323ID) and encrypting a challenge string presented by the gatekeeper. This assumes that the gatekeeper also knows the password associated with the subscription ID. The gatekeeper will authenticate the user by verifying that the challenge string was correctly encrypted.

The example registration procedure with gatekeeper authentication is as follows:

1) If the endpoint uses **GRQ** to discover a gatekeeper, one of the aliases in the message would be the subscription ID (as an **H323ID**). The **authenticationcapability** would contain an **AuthenticationMechanism** of **pwdSymEnc** and the **algorithmOIDs** would be set to indicate the entire set of encryption algorithms supported by the endpoint. (For example, one of these would be 56-bit DES in ECB mode.)

2) The gatekeeper would respond with **GCF** (assuming it recognizes the alias) carrying a **tokens** element containing one **ClearToken**. This **ClearToken** would contain both a **challenge** and a **timeStamp** element. The **challenge** would contain 16 octets. (To prevent replay attacks, the **ClearToken** should contain a **timeStamp**.) The **authenticationmode** should be set to **pwdSymEnc** and the **algorithmOID** should be set to indicate the encryption algorithm required by the gatekeeper (for example, 56-bit DES in ECB mode).

   If the gatekeeper does not support any of the **algorithmOIDs** indicated in the **GRQ**, then it would respond with a **GRJ** containing a **GatekeeperRejectReason** of **resourceUnavailable**.

3) The endpoint application should then attempt to register with (one of) the GK(s) that responded with a **GCF** by sending an **RRQ** containing a **cryptoEPPwdEncr** in the **cryptoTokens**. The **cryptoEPPwdEncr** would have the **algorithmOID** of the encryption algorithm agreed to in the **GRQ**/**GCF** exchange, and the encrypted challenge.

   The encryption key is constructed from the user's password using the procedure described in 8.2.1. The resulting octet "string" is then used as the DES key to encrypt the **challenge**.

4) When the gatekeeper receives the encrypted challenge in the **RRQ**, it would compare it to an identically generated encrypted challenge to authenticate the registering user. If the two encrypted strings do not match, the gatekeeper should respond with an **RRJ** with the **RegistrationRejectReason** set to **securityDenial** or other appropriate security error code according to clause 11.1. If they match, the gatekeeper sends an **RCF** to the endpoint.

5) If the gatekeeper receives an **RRQ** which does not contain an acceptable **cryptoTokens** element, then it should respond with an **RRJ** with a **GatekeeperRejectReason** of **discoveryRequired**. The endpoint, upon receiving such an **RRJ**, may perform discovery which will allow the gatekeeper/endpoint to exchange a new challenge.

   NOTE – The **GRQ** message may be unicast to the gatekeeper.

I.1.5 IPsec

In general, IPsec ([RFC 2401], RFC 2406 [ESP]) and RFC 2409 [IKE] can be used to provide authentication and, optionally, confidentiality (i.e., encryption) at the IP layer transparent to whatever (application) protocol runs above. The application protocol does not have to be updated to allow this; only security policy at each end.

For example, to make maximum use of IPsec for a simple point-to-point call, the following scenario could be followed:

1) The calling endpoint and its gatekeeper would set policy to require the use of IPsec (authentication and, optionally, confidentiality) on the RAS protocol. Thus, before the first RAS message is sent from the endpoint to the gatekeeper, the ISAKMP (RFC 2407)/Oakley (RFC 2412) daemon on the endpoint will negotiate security services to be used on packets to and from the RAS channel's well-known port. Once negotiation is complete, the RAS channel will operate exactly as if it were not secured. Using this secure channel the gatekeeper will inform the endpoint of the address and port number of the call signalling channel in the called endpoint.

2) After obtaining the address and port number of the call signalling channel, the calling endpoint would dynamically update its security policy to require the desired IPsec security on that address and protocol/port pair. Now, when the calling endpoint attempts to contact this address/port, the packets would be queued while an ISAKMP (RFC 2407)/Oakley (RFC 2412) negotiation is performed between the endpoints. Upon completion of this negotiation, an IPsec Security Association (SA) for the address/port will exist and the Q.931 signalling can proceed.

3) On the Q.931 SETUP and CONNECT exchange, the endpoints can negotiate the use of IPsec for the H.245 channel. This will allow the endpoints to again dynamically update their IPsec policy databases to force the use of IPsec on that connection.

4) As with the call signalling channel, a transparent ISAKMP (RFC 2407)/Oakley (RFC 2412) negotiation will take place before any H.245 packets are transmitted. The authentication performed by this ISAKMP (RFC 2407)/Oakley (RFC 2412) exchange will be the initial attempt at user-to-user authentication, and will set up a (probably) secure channel between the two users on which to negotiate the characteristics of the audio channel. If, after some person-to-person Q&A, either user is not satisfied with the authentication, different certificates can be chosen and the ISAKMP (RFC 2407)/Oakley (RFC 2412) exchange repeated.

5) After each H.245 ISAKMP (RFC 2407)/Oakley (RFC 2412) authentication, new keying material is exchanged for the RTP audio channel. This keying material is distributed by the master on the secure H.245 channel. Because the H.245 protocol is defined for the master to distribute the media keying material on the H.245 channel (to allow for multipoint communication), it is not recommended that IPsec be used for the RTP channel.

An encrypted H.245 channel is a potential problem for proxy or NAT firewall, since the dynamically-assigned port numbers are carried in the H.245 protocol. Such firewalls would have to decipher, modify and re-encipher the protocol to operate correctly. For this reason, the "Security" Logical Channel was introduced into ITU-T Rec. H.245. If this channel is used, the H.245 channel can remain unsecured; authentication and key-generation would be done with the "Security" Logical Channel. Logical channel signalling would allow this channel to be protected with IPsec, and the secret key used on the "Security" Logical Channel would be used to protect the **EncryptionSync** distributed by the master on the H.245 channel.

I.1.6 Back-end service support

Back-end servers are an important supplementary function in an overall H.323-based multimedia environment. For example, BES provides services for user authentication, for service authorization, also for accounting, charging and billing and other services. In a simple model, the gatekeeper could provide such services. In a decomposed architecture, the GK may not always provide such services; either because it may not have access to the BES databases, or it may be part of a different administrative domain. Likewise, the terminal or user usually does not know their BES.

Figure I.2 shows a scenario with a multimedia terminal (e.g., a SASET), a gatekeeper and linked BES. It is not within the scope of ITU-T Rec. H.323 as to how exactly the BES communicates with the GK. Several methods and protocols could be applicable: RADIUS (see RFC 2865) is considered as one of the most important ones, which is widely deployed by service providers.

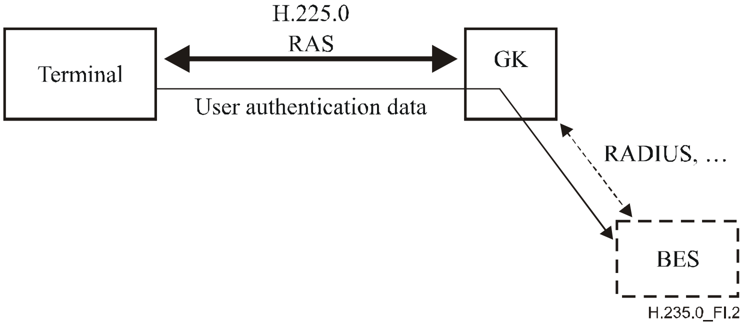


Figure I.2/H.235.0 – Scenario with back-end server

A GK offering BES support should support at least the following two modes:

1) **default mode**: in this mode, the terminal does not know the BES, and requires a trust relationship with the GK. The terminal sends the user authentication data in encrypted form (**cryptoEncryptedToken**) to the GK, which decrypts it, extracts the user authentication information and applies it towards the BES. The password-based encryption of the **ClearToken** is accomplished by applying a distinct secret that is shared between the terminal and the GK to the **CryptoToken**. The encryption key could be derived from the password with which the terminal securely registers at the GK.

   CryptoToken carries **cryptoEncryptedToken** where **tokenOID** is set to "M" indicating BES default mode; and **token** holding:

   - **algorithmOID** indicating the encryption algorithm; "Y" (DES56-CBC), "Z" (3DES-OCBC); see clause 11/H.235.6;
   - **paramS** unused;
   - **encryptedData** set to the octet representation of the encrypted **ClearToken**.

   The **ClearToken** holds as **password** the user authentication data. Protected **ClearToken** information could be password/PIN, user identification, prepaid calling card number and credit card number. The **timestamp** is set to the current time of the terminal, **random** contains a monotonically increasing sequence number, **sendersID** is set to the terminal ID and **generalID** to the GK identifier. The initial value of the encryption algorithm shall be kept constant; it could be part of the terminal subscription secret.

   NOTE – The **ClearToken** is not transmitted.

2) **RADIUS mode**: in this mode, the BES and the terminal user share a common secret and the GK should not be trusted for the BES RADIUS authentication. The GK simply forwards a RADIUS challenge received from the BES within *Access-Challenge* towards the terminal and sends the user's response as a RADIUS response within *Access-Request* in the reverse direction. Terminal and GK negotiate this **radius** challenge/response capability in **AuthenticationBES** within **AuthenticationMechanism** during gatekeeper discovery.

   Upon receipt of a RADIUS *Access-Challenge* message conveying a challenge, the GK puts the 16-octet challenge in the **challenge** field of the **ClearToken** when querying the terminal with a **GCF** or any other RAS message. The **tokenOID** 'K' in the **ClearToken** indicates a RADIUS challenge.

The terminal may then present the challenge to the user and wait for the response entered. The terminal shall reply with a RAS message where the response is put into the **challenge** field of the **ClearToken**. The **tokenOID** 'L' in the **ClearToken** indicates a RADIUS response.

Table I.1 lists all the referenced OIDs.

Table I.1/H.235.0 – Object identifiers used by I.1.6

Object identifier reference	Object identifier value	Description
"K"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 31}	indicates a RADIUS challenge in the ClearToken
"L"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 32}	indicates a RADIUS response (conveyed in the challenge field) in the ClearToken
"M"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 33}	indicates BES default mode with a protected password in the ClearToken

Appendix II

H.324 implementation details

For further study.

Appendix III

Other H-series implementation details

For further study.

Appendix IV

Section mapping of H.235v3Amd1Cor1 to H.235v4 subseries Recommendations

This informative appendix shows the placement of all the sections of H.235v3Amd1Cor1 within the H.235v4 subseries Recommendations.

Table IV.1/H.235.0 – Clause mapping

H.235v3Amd1Cor1 Clause	Title	H.235v4.x subseries Recommendation	Clause
Main body	–	–	–
1	Scope	H.235.0	1
2	References	H.235.0 H.235.1 H.235.2 H.235.3	2 2 2 2
3	Terms and definitions	H.235.0 H.235.2 H.235.6	3 3 3
4	Symbols and abbreviations	H.235.0 H.235.3 H.235.6	4 4 4
5	Conventions	H.235.0 H.235.2 H.235.6	5 5 5
6	System introduction	H.235.0	6
6.1	Summary	H.235.0	6.1
6.2	Authentication	H.235.0	6.2
6.2.1	Certificates	H.235.0	6.2.1
6.3	Call establishment security	H.235.0	6.3
6.4	Call control (H.245) security	H.235.0	6.4
6.5	Media stream privacy	H.235.0	6.5
6.6	Trusted elements	H.235.0	6.6
6.6.1	Key escrow	H.235.0	6.6.1
6.7	Non-repudiation	H.235.0	6.7
6.8	Mobility security	H.235.0	6.8
6.9	Security profiles	H.235.0	6.9
7	Connection establishment procedures	H.235.0	7
7.1	Introduction	H.235.0	–
8	H.245 signalling and procedures	H.235.6	7

Table IV.1/H.235.0 – Clause mapping

H.235v3Amd1Cor1 Clause	Title	H.235v4.x subseries Recommendation	Clause
8.1	Secure H.245 channel operation	H.235.6	7.1
8.2	Unsecured H.245 channel operation	H.235.6	7.2
8.3	Capability exchange	H.235.6	7.3
8.4	Master role	H.235.6	7.4
8.5	Logical channel signalling	H.235.6	7.5
8.6	Fast connect security	H.235.6	7.6
8.6.1	Unidirectional fast start security	H.235.6	7.6.1
8.6.1.1	Using multiple encryption algorithms in fast connect	H.235.6	7.6.1.1
8.6.2	Bidirectional fast start security	H.235.6	7.6.2
8.7	Encrypted H.245 DTMF	H.235.6	7.7
8.7.1	Encrypted basic string	H.235.6	7.7.1
8.7.2	Encrypted iA5 string	H.235.6	7.7.2
8.7.3	Encrypted general string	H.235.6	7.7.3
8.7.4	List of object identifiers	H.235.6	7.7.4
8.8	Diffie-Hellman operation	H.235.6	7.8
9	Multipoint procedures	H.235.6	8.8
9.1	Authentication	H.235.6	8.8.1
9.2	Privacy	H.235.6	8.8.2
10	Authentication signalling and procedures	H.235.0	8
10.1	Introduction	H.235.0	---
10.2	Diffie-Hellman with optional authentication	H.235.0	8.1
10.3	Subscription-based authentication	H.235.0	8.2
10.3.1	Introduction	H.235.0	–
10.3.2	Password with symmetric encryption	H.235.0	8.2.1
10.3.3	Password with hashing	H.235.0	8.2.2
10.3.4	Certificate-based with signatures	H.235.0	8.2.3
10.3.5	Usage of shared secret and passwords	H.235.0	8.2.4
11	Media stream encryption procedures	H.235.6	9
11.1	Media session keys	H.235.6	9.1
11.2	Media anti-spamming	H.235.6	9.2
11.2.1	List of object identifiers	H.235.6	9.2.1
12	Security error recovery	H.235.0	11
13	Asymmetric authentication and key exchange using elliptic curve crypto systems	H.235.0	9
13.1	Key management	H.235.0	9.1

Table IV.1/H.235.0 – Clause mapping

H.235v3Amd1Cor1 Clause	Title	H.235v4.x subseries Recommendation	Clause
13.2	Digital signature	H.235.0	9.2
Appendix I	H.323 implementation details	H.235.0	Appendix I
I.1	Ciphertext padding methods	H.235.6	I.1
I.2	News keys	H.235.6	8.7.2
I.3	H.323 trusted elements	H.235.6	8.7.3
I.4	Implementation examples	H.235.0	I.1
I.4.1	Tokens	H.235.0	I.1.1
I.4.2	Token usage in H.323 systems	H.235.0	I.1.2
I.4.3	H.235 random value usage in H.323 systems	H.235.0	I.1.3
I.4.4	Password	H.235.0	I.1.4
I.4.5	IPsec	H.235.0	I.1.5
I.4.6	Back-end service support	H.235.0	I.1.6
Appendix II	H.324 implementation details	H.235.0	Appendix II
Appendix III	Other H-series implementation details	H.235.0	Appendix III
Appendix IV	Bibliography	H.235.0	2.2
Annex A	**H.235 ASN.1**	**H.235.0**	**Annex A**
Annex B	**H.323 specific topics**	**H.235.6**	–
B.1	Background	H.235.0	6
B.2	Signalling and procedures	H.235.6	8
B.2.1	Revision 1 compatibility	H.235.6	8.1
B.2.2	Error signalling	H.235.0	11.1
B.2.3	Version 3 feature indication	H.235.6	8.2
B.2.4	Key transport	H.235.6	8.3
B.2.4.1	Improved key transport in H.235 version 3	H.235.6	8.3.1
B.2.5	Enhanced OFB mode	H.235.6	8.4
B.2.6	Key update and synchronization	H.235.6	8.6
B.2.6.1	Unacknowledged key update	H.235.6	8.6.1
B.2.6.2	Improved key update	H.235.6	8.6.2
B.2.6.3	Payload-type-based key update and synchronization	H.235.6	8.6.3
B.3	RTP/RTCP issues	H.235.6	9.3
B.3.1	Initialization vectors	H.235.6	9.3.1
B.3.1.1	CBC initialization vector	H.235.6	9.3.1.1
B.3.1.2	EOFB initialization vector	H.235.6	9.3.1.2
B.3.2	Padding	H.235.6	9.3.2
B.3.3	RTCP protection	H.235.6	9.3.3

Table IV.1/H.235.0 – Clause mapping

H.235v3Amd1Cor1 Clause	Title	H.235v4.x subseries Recommendation	Clause
B.3.4	Secured payload stream	H.235.6	9.3.4
B.3.5	Interworking with J.170	H.235.6	9.3.5
B.4	RAS signalling/procedures for authentication	H.235.0	8.3
B.4.1	Introduction	H.235.0	–
B.4.2	Endpoint-gatekeeper authentication (non-subscription-based)	H.235.0	8.3.1
B.4.3	Endpoint-gatekeeper authentication (subscription-based)	H.235.0	8.3.2
B.4.3.1	Password with symmetric encryption	H.235.0	8.3.2.1
B.4.3.2	Password with hashing	H.235.0	8.3.2.2
B.4.3.3	Certificate-based with signatures	H.235.0	8.3.3.3
B.5	Non-terminal interactions	H.235.6	8.7
B.5.1	Gateway	H.235.6	8.7.1
B.6	Key management on the RAS channel	H.235.0	8.4
B.7	Pseudo-Random Function (PRF)	H.235.0	10
Annex C	**H.324-specific topics**	**H.235.0**	**Annex B**
Annex D	**Baseline security profile**	**H.235.1**	
D.1	Introduction	H.235.1	
D.2	Conventions	H.235.1	5
D.3	Scope	H.235.1	1
D.4	Abbreviations	H.235.1	4
D.5	Normative references	H.235.1	2.1
D.6	Baseline security profile	H.235.1	
D.6.1	Overview	H.235.1	6.1
D.6.1.1	Baseline security profile	H.235.1	6.2
D.6.1.2	Voice encryption security profile	H.235.6	6.1
D.6.2	Authentication and integrity	H.235.1	3.1
D.6.3	H.323 requirements	H.235.1	6.3
D.6.3.1	Overview	H.235.1	6.4
D.6.3.2	Symmetric-key-based signalling message authentication details (Procedure I)	H.235.1	7
D.6.3.3	Computation of the password-based hash	H.235.1	7.1
D.6.3.3.1	HMAC-SHA1-96	H.235.1	7.2
D.6.3.3.2	Authentication and integrity	H.235.1	7.3
D.6.3.3.3	Authentication-only (Procedure IA)	H.235.1	8
D.6.3.4	Usage illustration for Procedure I	H.235.1	9

Table IV.1/H.235.0 – Clause mapping

H.235v3Amd1Cor1 Clause	Title	H.235v4.x subseries Recommendation	Clause
D.6.3.4.1	RAS message authentication and integrity	H.235.1	9.1
D.6.3.4.2	H.225.0 message authentication and integrity	H.235.1	9.2
D.6.3.4.3	H.245 message authentication and integrity	H.235.1	9.3
D.6.4	Direct-routed scenario	H.235.1	9.4
D.6.5	Back-end-service support	H.235.1	10
D.6.6	H.235 version 1 compatibility	H.235.1	11
D.6.7	Multicast behaviour	H.235.1	12
D.7	Voice encryption security profile	H.235.6	6.1
D.7.1	Key management	H.235.6	8.5
D.7.2	Key update and synchronization	H.235.6	8.6
D.7.3	Triple DES in outer CBC mode	H.235.6	9.4
D.7.4	DES algorithm operating in EOFB mode	H.235.6	9.5
D.7.5	Triple DES in outer EOFB mode	H.235.6	9.6
D.8	Lawful interception	H.235.6	10
D.9	List of secured signalling messages	H.235.1	13
D.9.1	H.225.0 RAS	H.235.1	13.1
D.9.2	H.225.0 call signalling	H.235.1	13.2
D.9.3	H.245 call control	H.235.1	13.3
D.10	Usage of sendersID and generalID	H.235.1	14
D.11	List of object identifiers	H.235.1 H.235.6	15 11
D.12	Bibliography	H.235.1 H.235.6	2.2 2.2
Annex E	**Signature security profile**	**H.235.2**	
E.1	Overview	H.235.2	6
E.2	Specification conventions	H.235.2	5
E.3	H.323 requirements	H.235.2	6.1
E.4	Security services	H.235.2	5
E.5	Digital signatures with public/private key pairs details (Procedure II)	H.235.2	7
E.6	Multipoint conferencing procedures	H.235.2	8
E.7	End-to-end authentication (Procedure III)	H.235.2	9
E.8	Authentication-only	H.235.2	10
E.9	Authentication and integrity	H.235.2	11
E.10	Computation of the digital signature	H.235.2	12

Table IV.1/H.235.0 – Clause mapping

H.235v3Amd1Cor1 Clause	Title	H.235v4.x subseries Recommendation	Clause
E.11	Verification of the digital signature	H.235.2	13
E.12	Handling of certificates	H.235.2	14
E.13	Usage illustration for Procedure II	H.235.2	15
E.13.1	RAS message authentication, integrity and non-repudiation	H.235.2	15.1
E.13.2	RAS authentication only	H.235.2	15.2
E.13.3	H.225.0 message authentication, integrity and non-repudiation	H.235.2	15.3
E.13.4	H.245 message authentication and integrity	H.235.2	15.4
E.14	H.235 version 1 compatibility	H.235.2	16
E.15	Multicast behaviour	H.235.2	17
E.16	List of secure signalling messages	H.235.2	18
E.16.1	H.225.0 RAS	H.235.2	18.1
E.16.2	H.225.0 call signalling	H.235.2	18.2
E.17	Usage of sendersID and generalID	H.235.2	19
E.18	List of object identifiers	H.235.2	20
Appendix IV (Annex E)	Bibliography	H.235.2	2.2
Annex F	**Hybrid security profile**	**H.235.3**	
F.1	Overview	H.235.3	6
F.2	Normative references	H.235.3	2.1
F.3	Acronyms	H.235.3	4
F.4	Specification conventions	H.235.3	5
F.5	H.323 requirements	H.235.3	6.1
F.6	Authentication and integrity	H.235.3	6.2
F.7	Procedure IV	H.235.3	7
F.8	Security association for concurrent calls	H.235.3	8
F.9	Key update	H.235.3	9
F.10	Illustration examples	H.235.3	11
F.11	Multicast behaviour	H.235.3	12
F.12	List of secure signalling messages	H.235.3	13
F.12.1	H.225.0 RAS	H.235.3	13.1
F.12.2	H.225.0 call signalling (single administrative domain)	H.235.3	13.2
F.12.3	H.225.0 call signalling (multi-administrative domain)	H.235.3	13.3
F.13	List of object identifiers	H.235.3	14
Appendix IV	Bibliography	H.235.3	2.2

Table IV.1/H.235.0 – Clause mapping

H.235v3Amd1Cor1 Clause	Title	H.235v4.x subseries Recommendation	Clause
Annex G	**Usage of the Secure Real-Time Transport Protocol (SRTP) in conjunction with the MIKEY key management protocol within H.235**	**H.235.7**	
G.1	Scope	H.235.7	1
G.2	References	H.235.7	2
G.2.1	Normative References	H.235.7	2.1
G.2.2	Informative references	H.235.7	2.2
G.3	Terms and Definitions	H.235.7	3
G.4	Symbols and Abbreviations	H.235.7	4
G.5	Specification Conventions	H.235.7	5
G.6	Introduction	H.235.7	6
G.7	Overview and Scenarios	H.235.7	7
G.7.1	MIKEY operation at "session level"	H.235.7	7.1
G.7.2	MIKEY operation at "media level"	H.235.7	7.2
G.7.3	MIKEY Capability Negotiation	H.235.7	7.3
G.8	Security Profile using Symmetric Security Techniques	H.235.7	8
G.8.1	Terminating a H.323 Call	H.235.7	8.1
G.8.2	TGK re-keying and CSB updating	H.235.7	8.2
G.8.3	H.245 tunnelling support	H.235.7	8.3
G.8.4	SRTP algorithms	H.235.7	8.4
G.8.5	List of Object Identifiers	H.235.7	8.5
G.9	Security Profile using Asymmetric Security Techniques	H.235.7	9
G.9.1	Terminating a H.323 Call	H.235.7	9.1
G.9.2	TGK re-keying and CSB updating	H.235.7	9.2
G.9.3	H.245 tunnelling support	H.235.7	9.3
G.9.4	SRTP algorithms	H.235.7	9.4
G.9.5	List of Object Identifiers	H.235.7	9.5
G.I	MIKEY-DHHMAC option	H.235.7	Appendix I
G.I.1	Terminating a H.323 Call	H.235.7	I.1
G.I.2	TGK re-keying and CSB updating	H.235.7	I.2
G.II	Using H.235 Annex I for establishing a pre-shared secret	H.235.7	Appendix II
G.II.1	Terminating a H.323 Call	H.235.7	II.1
G.II.2	TGK re-keying and CSB updating	H.235.7	II.2

Table IV.1/H.235.0 – Clause mapping

H.235v3Amd1Cor1 Clause	Title	H.235v4.x subseries Recommendation	Clause
Annex H	**RAS key management**	**H.235.5**	
H.1	Introduction	H.235.5	–
H.2	Scope	H.235.5	1
H.3	References	H.235.5	2
H.3.1	Normative references	H.235.5	2.1
H.3.2	Informative references	H.235.5	2.2
H.4	Definitions	H.235.5	3
H.5	Abbreviations	H.235.5	4
H.6	Basic framework	H.235.5	6
H.6.1	Improved negotiation capabilities in H.235v3	H.235.5	6.1
H.6.2	Use between endpoint and gatekeeper	H.235.5	6.2
H.6.3	Use of profile between gatekeepers	H.235.5	6.3
H.6.4	Signalling channel encryption and authentication	H.235.5	6.4
H.7	A specific security profile (SP1)	H.235.5	7
H.8	Extensions to the framework (Informative)	H.235.5	9
H.8.1	Using the master key to secure the call signalling channel via TLS	H.235.5	9.1
H.8.1.1	Endpoint registration	H.235.5	9.1.1
H.8.2	Use of certificates to authenticate the gatekeeper	H.235.5	9.2
H.8.3	Use of alternative signalling security mechanisms	H.235.5	9.3
H.9	Threats (Informative)	H.235.5	10
H.9.1	Passive attack	H.235.5	10.1
H.9.2	Denial-of-Service attacks	H.235.5	10.2
H.9.3	Man-in-the-Middle attacks	H.235.5	10.3
H.9.4	Guessing attacks	H.235.5	10.4
H.9.5	Unencrypted gatekeeper half-key	H.235.5	10.5
Annex I	**Support of direct-routed calls**	**H.235.4**	
I.1	Scope	H.235.4	1
I.2	Introduction	H.235.4	6
I.3	Specification conventions	H.235.4	5
I.4	Terms and definitions	H.235.4	3
I.5	Symbols and abbreviations	H.235.4	4
I.6	Normative references	H.235.4	2
I.7	Overview	H.235.4	7

Table IV.1/H.235.0 – Clause mapping

H.235v3Amd1Cor1 Clause	Title	H.235v4.x subseries Recommendation	Clause
I.8	Limitations	H.235.4	8
I.9	Procedure DRC	H.235.4	9
I.10	PRF-based key derivation procedure	H.235.4	12
I.11	FIPS-140 based key derivation procedure	H.235.4	13
I.12	List of object identifiers	H.235.4	14
Appendix I (Annex I)	Bibliography	H.235.4	2.2

Appendix V

Figure mapping of H.235v3Amd1Cor1 to H.235v4 subseries Recommendations

This informative appendix shows the placement of all the Figures of H.235v3Amd1Cor1 within the H.235v4 subseries Recommendations.

Table V.1/H.235.0 – Figure mapping

H.235v3Amd1Cor1 Figure	Title	H.235v4.x subseries Recommendation	Figure
Figure 1	Diffie-Hellman with optional authentication	H.235.0	4
Figure 2a	Password with symmetric encryption; two passes	H.235.0	5
Figure 2b	Password with symmetric encryption; three passes	H.235.0	6
Figure 3a	Password with hashing; two passes	H.235.0	7
Figure 3b	Password with hashing; three passes	H.230.0	8
Figure 4a	Certificate-based with signatures; two passes	H.235.0	9
Figure 4b	Certificate-based with signatures; three passes	H.235.0	10
Figure 5	Encryption of media	H.235.6	7
Figure 6	Decryption of media	H.235.6	8
Figure 7	RTP packet format for media anti-spamming	H.235.6	9
Figure I.1	Ciphertext stealing in ECB mode	H.235.6	I.1
Figure I.2	Ciphertext stealing in CBC mode	H.235.6	I.2
Figure I.2a	Zero padding in CBC mode	H.235.6	I.3
Figure I.3	Zero padding in CFB mode	H.235.6	I.4
Figure I.4	Zero padding in OFB mode	H.235.6	I.5

Table V.1/H.235.0 – Figure mapping

H.235v3Amd1Cor1 Figure	Title	H.235v4.x subseries Recommendation	Figure
Figure I.4.1	EOFB mode with zero padding	H.235.6	I.6
Figure I.5	Padding as prescribed by RTP	H.235.6	I.7
Figure I.6	Tokens	H.235.0	I.1
Figure I.7	Scenario with back-end server	H.235.0	I.2
Figure B.1	Overview	H.235.0	2
Figure B.1.1	Unacknowledged session key distribution/key update from the master to the slave(s)	H.235.6	4
Figure B.1.2	Session key update on slave's logical channel	H.235.6	5
Figure B.1.3	Session key update on master's logical channel	H.235.6	6
Figure B.2	Password with symmetric encryption	H.235.0	11
Figure B.3	Password with hashing	H.235.0	12
Figure B.4	Certificate-based with signatures	H.235.0	13
Figure D.1	Illustrating procedure I usage in a GK-GK scenario with both EPs in GK-routed zones	H.235.1	1
Figure D.2	Illustrating procedure I usage in a mixed scenario with EP1 in a GK-routed zone and EP2 in a direct-routed zone	H.235.1	2
Figure D.3	Illustrating procedure I usage in a scenario with both EPs in zones using a direct-routed GK	H.235.1	3
Figure D.4	Triple-DES encryption in outer CBC mode	H.235.6	10
Figure D.5	Triple-DES encryption in outer EOFB mode	H.235.6	11
Figure E.1	Simultaneous use of hop-by-hop security and end-to-end authentication	H.235.2	1
Figure E.2	Illustrating public-key usage in a GK-GK routed model	H.235.2	2
Figure F.1	Security association for concurrent calls	H.235.3	1
Figure F.2	Flow diagram in a single administrative domain	H.235.3	2
Figure F.3	Flow diagram in a multi-administrative domain	H.235.3	3
Figure G.1	Scenario	H.235.7	1
Figure G.2	Security scenario with MIKEY and SRTP	H.235.7	2
Figure G.3	Hop-by-Hop scenario only with shared secrets	H.235.7	3

Table V.1/H.235.0 – Figure mapping

H.235v3Amd1Cor1 Figure	Title	H.235v4.x subseries Recommendation	Figure
Figure G.4	Example Endpoint B calling Endpoint A (GK-routed) with MIKEY-preshared	H.235.7	4
Figure G.5	MIKEY-preshared processing by EP B	H.235.7	5
Figure G.6	MIKEY-preshared processing by EP A	H.235.7	6
Figure G.7	Example Endpoint B terminates a call	H.235.7	7
Figure G.8	Example Endpoint B updating a key	H.237.7	8
Figure G.9	End-to-end scenario using PKI (multiple GKs)	H.235.7	9
Figure G.10	Example EP B calls EP A (multiple GK-routed) with MIKEY-PK-SIGN	H.235.7	10
Figure G.11	MIKEY-PK-SIGN processing by EP B	H.235.7	11
Figure G.12	MIKEY-PK-SIGN processing by EP A	H.235.7	12
Figure G.13	Example Endpoint B terminates a call	H.235.7	13
Figure G.14	Example EP B (Initiator) initiated TGK re-keying and key update	H.235.7	14
Figure G.I-1	Example Endpoint B calling Endpoint A (GK-routed) with MIKEY-DHHMAC	H.235.7	I.1
Figure G.I-2	Example Endpoint B terminates a call	H.235.7	I.2
Figure G.I-3	Example Endpoint B updating a key	H.235.7	I.3
Figure G.II-1	Example Endpoint B calling Endpoint A (non-GK-routed) with MIKEY-Preshared and H.235.4 DRC1	H.235.7	II.1
Figure H.1	Information flow for security profile and TLS	H.235.5	1
Figure I.1	Direct-routed call scenario	H.235.4	1
Figure I.2	Basic communication flow	H.235.4	2

Appendix VI

Table mapping of H.235v3Amd1Cor1 to H.235v4 subseries Recommendations

This informative appendix shows the placement of all the Tables of H.235v3Amd1Cor1 within the H.235v4 subseries Recommendations.

Table VI.1/H.235.0 – Table mapping

H.235v3Amd1Cor1 Table	Title	H.235v4.x subseries Recommendation	Table
Table 1	Object identifier for NULL encryption	H.235.6	2
Table 2	Object identifiers for H.245 DTMF encryption	H.235.6	3
Table 3	Object identifiers used for anti-spamming	H.235.6	5
Table I.1	Object identifiers used by I.4.6	H.235.0	I.1
Table D.1	Summary of Annex D security profiles	----	---
Table D.2	Baseline security profile	H.235.1	1
Table D.3	Voice encryption profile	H.235.6	1
Table D.4	Diffie-Hellman groups	H.235.6	4
Table D.5	Usage of sendersID and generalID	H.235.1	2
Table D.6	Object identifiers used by Annex D	H.235.1 H.235.6	3 6
Table E.1	Signature security profile	H.235.2	1
Table E.2	Usage of sendersID and GeneralID	H.235.2	2
Table E.3	Object identifiers used by Annex E	H.235.2	3
Table F.1	Overview of the hybrid security profile	H.235.3	1
Table F.2	Object identifiers used by Annex F	H.235.3	2
Table G.1	MIKEY Key Management Protocols	H.235.7	1
Table H.1	Profile elements	H.235.5	1
Table I.0	Calculating encryption and salting keys from a shared secret	H.235.4	1
Table I.1	Object identifiers used by H.235.4	H.235.4	2

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2006